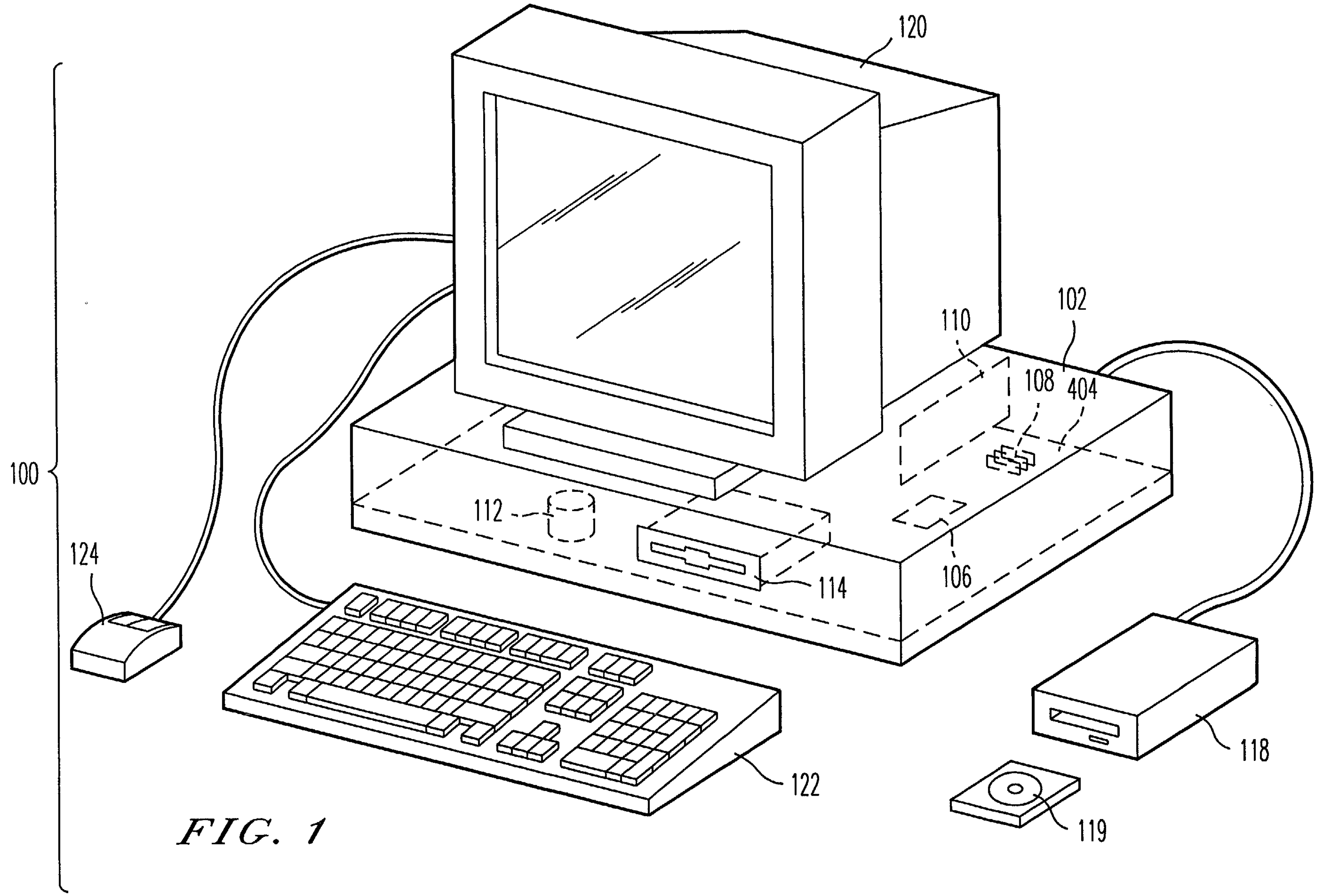


FIG. 1

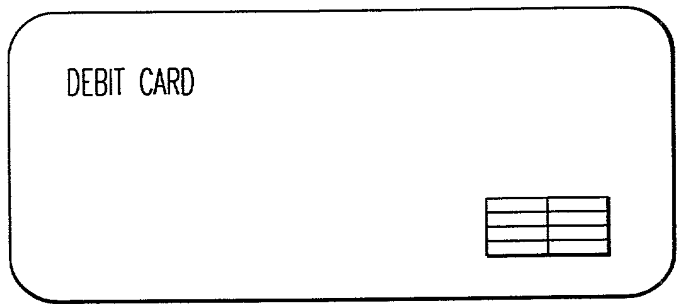


FIG. 2

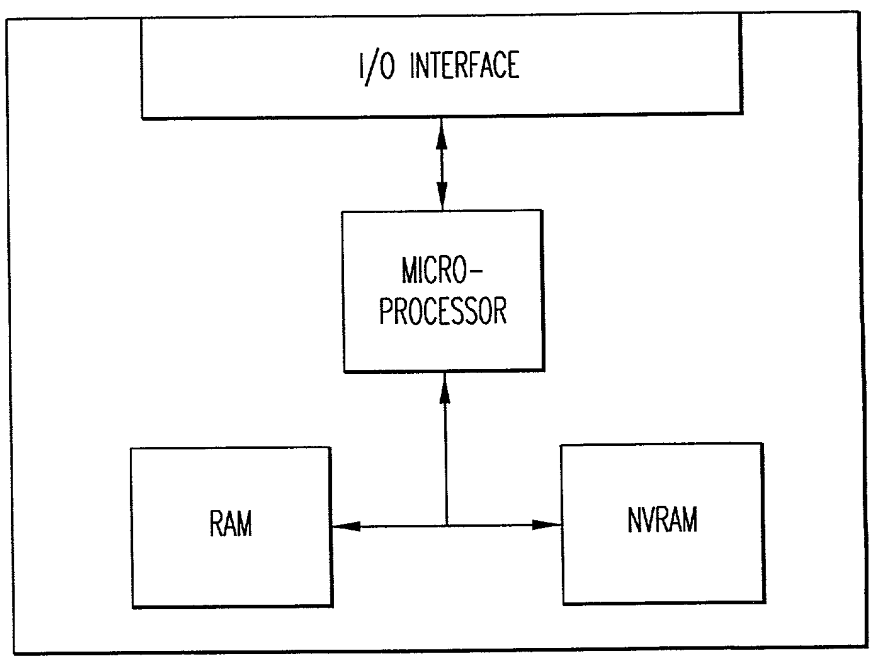


FIG. 3

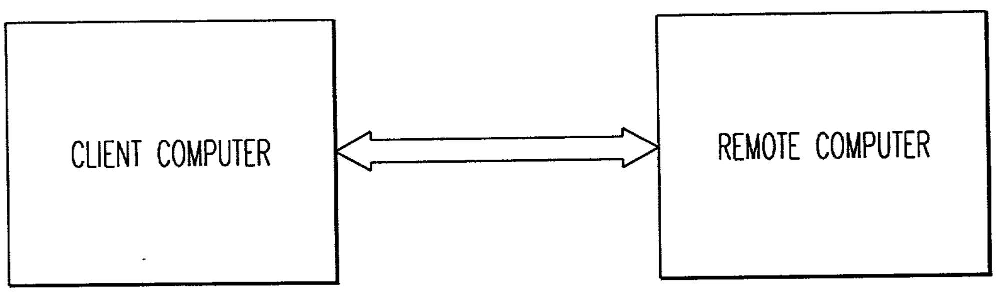


FIG. 4

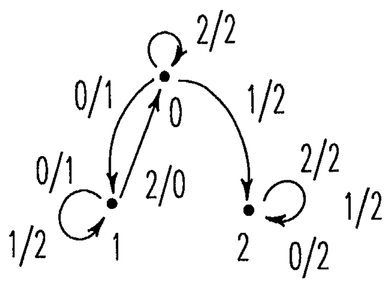


FIG. 5A

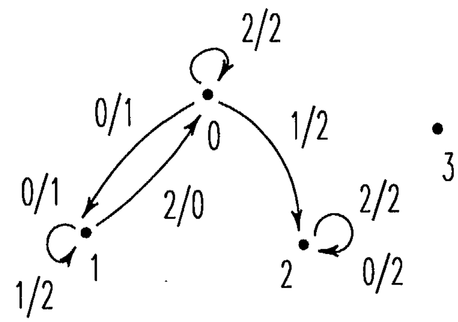


FIG. 5C

STATE \ INPUT	0	1	2
0	(1,1)	(2,2)	(0,2)
1	(1,1)	(1,2)	(0,0)
2	(2,2)	(2,2)	(2,2)

FIG. 5B

STATE \ INPUT	0	1	2
0	(1,1)	(2,2)	(0,2)
1	(1,1)	(1,2)	(0,0)
2	(2,2)	—	(2,2)

FIG. 5D

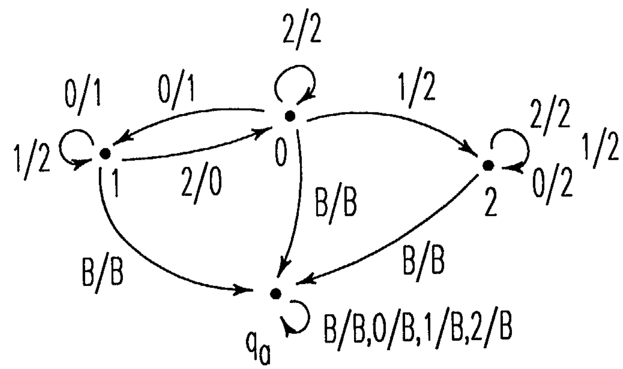


FIG. 6A

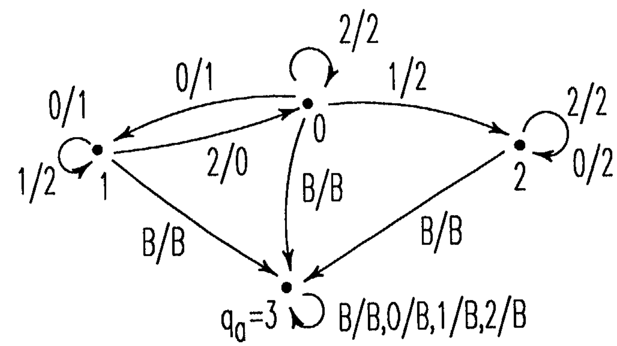


FIG. 6C

CORRESPONDING FUNCTION TABLE

STATE \ INPUT	0	1	2	B
0	(1,1)	(2,2)	(0,2)	(q_a,B)
1	(1,1)	(1,2)	(0,0)	(q_a,B)
2	(2,2)	(2,2)	(2,2)	(q_a,B)
q_a	(q_a,B)	(q_a,B)	(q_a,B)	(q_a,B)

FIG. 6B

CORRESPONDING FUNCTION TABLE

STATE \ INPUT	0	1	2	B
0	(1,1)	(2,2)	(0,2)	(3,B)
1	(1,1)	(1,2)	(0,0)	(3,B)
2	(2,2)	—	(2,2)	(3,B)
q_a=3	(3,B)	(3,B)	(3,B)	(3,B)

FIG. 6D

INPUT SPACE: $\Sigma' = \{0,1,2,B\}$

STATE SPACE: $Q' = \{0,1,2,q_a\},\ q_a=3$

OUTPUT SPACE: $\Delta' = \{0,1,2,3\}$

VECTORIZATION EXAMPLE FOR N=2:

INPUT SPACE: $\Sigma' = \{\overset{0}{(0,0)},\ \overset{1}{(0,1)},\ \overset{2}{(1,0)},\ \overset{B}{(1,1)}\}$

STATE SPACE: $Q' = \{\overset{0}{(0,0)},\ \overset{1}{(0,1)},\ \overset{2}{(1,0)},\ \overset{q_a=3}{(1,1)}\}$

OUTPUT SPACE: $\Delta' = \{\overset{0}{(0,0)},\ \overset{1}{(0,1)},\ \overset{2}{(1,0)},\ \overset{3}{(1,1)}\}$

FIG. 7A

VECTORIZATION EXAMPLE FOR N=3:

INPUT SPACE: $\Sigma' = \{\overset{0}{(0,0)},\ \overset{1}{(0,1)},\ \overset{2}{(0,2)},\ \overset{B}{(1,0)}\}$

STATE SPACE: $Q' = \{\overset{0}{(0,0)},\ \overset{1}{(0,1)},\ \overset{2}{(0,2)},\ \overset{q_a=3}{(1,0)}\}$

OUTPUT SPACE: $\Delta' = \{\overset{0}{(0,0)},\ \overset{1}{(0,1)},\ \overset{2}{(0,2)},\ \overset{B}{(1,0)}\}$

FIG. 7B

VECTORIZATION EXAMPLE FOR $N \geq 4$

INPUT SPACE: $\Sigma' = \{\overset{0}{(0)},\ \overset{1}{(1)},\ \overset{2}{(2)},\ \overset{B}{(3)}\}$

STATE SPACE: $Q' = \{\overset{0}{(0)},\ \overset{1}{(1)},\ \overset{2}{(2)},\ \overset{q_a=3}{(3)}\}$

OUTPUT SPACE: $\Delta' = \{\overset{0}{(0)},\ \overset{1}{(1)},\ \overset{2}{(2)},\ (3)\}$

FIG. 7C

VECTORIZATION EXAMPLE FOR $N'=2$:

INPUT SPACE: $\Sigma' = \{(0,0), (0,1), (1,0), (1,1)\}$

STATE SPACE: $Q' = \{(0,0), (0,1), (1,0), (1,1)\}$

OUTPUT: $\Delta' = \{(0,0), (0,1), (1,0), (1,1)\}$

IN THIS CASE N MAY BE SET TO ANY PRIME NUMBER ≥ 2.
SELECTING PRIMES $N>2$ RESULTS IN $(N-2)^2$ INPUT, STATE AND OUTPUT REPRESENTATIONS THAT INITIALLY REMAIN UNUSED.

FIG. 8A

VECTORIZATION EXAMPLE FOR $N'=3$:

INPUT SPACE: $\Sigma' = \{(0,0), (0,1), (0,2), (1,0)\}$

STATE SPACE: $Q' = \{(0,0), (0,1), (0,2), (1,0)\}$

OUTPUT: $\Delta' = \{(0,0), (0,1), (0,2), (1,0)\}$

IN THIS CASE N MAY BE SET TO ANY PRIME NUMBER ≥ 3.
FOR EVERY N THERE ARE N^2-4 UNUSED REPRESENTATIONS FOR INPUT VECTORS (INPUT "SYMBOLS"), STATE VECTORS, AND OUTPUT VECTORS (OUTPUT "SYMBOLS").

FIG. 8B

VECTORIZATION EXAMPLE FOR $N' \geq 4$:

INPUT SPACE: $\Sigma' = \{(0), (1), (2), (3)\}$

STATE SPACE: $Q' = \{(0), (1), (2), (3)\}$

OUTPUT: $\Delta' = \{(0), (1), (2), (3)\}$

IN THIS CASE N MAY BE SET TO ANY PRIME NUMBER ≥ 5
FOR EVERY N THERE ARE $N-4$ UNUSED REPRESENTATIONS FOR INPUT VECTORS, STATE VECTORS, AND OUTPUT VECTORS.

SELECTING AN N SUCH THAT THERE ARE MORE VALUES FOR N THAN OUTPUT VECTORS, INPUT VECTORS OR STATES IS SOMETHING THAT CAN BE DONE TO INCREASE THE POSSIBILITIES FOR INTRODUCING RANDOMNESS INTO THE PLAINTEXT STATES MACHINE

FIG. 8C

	STATE \ INPUT	(0,0)	(0,1)	(0,2)	(1,0)
0	(0,0)	((0,1),(0,1))	((0,2),(0,2))	((0,0),(0,2))	((1,0),(1,0))
1	(0,1)	((0,1),(0,1))	((0,1),(0,2))	((0,0),(0,0))	((1,0),(1,0))
2	(0,2)	((0,2),(0,2))	——	((0,2),(0,2))	((1,0),(1,0))
q_a = 3	(1,0)	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
4	(1,1)	——	——	——	——
5	(1,2)	——	——	——	——
6	(2,0)	——	——	——	——
7	(2,1)	——	——	——	——
8	(2,2)	——	——	——	——

FIG. 9A

STATE \ INPUT	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)	(2,0)	(2,1)	(2,2)
(0,0)	((0,1),(0,1))	((0,2),(0,2))	((0,0),(0,2))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
(0,1)	((0,1),(0,1))	((0,1),(0,2))	((0,0),(0,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
(0,2)	((0,2),(0,2))	((1,0),(1,0))	((0,2),(0,2))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
(1,0)	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
(1,1)	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
(1,2)	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
(2,0)	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
(2,1)	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
(2,2)	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))

FIG. 9B

STATE \ INPUT	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)	(2,0)	(2,1)	(2,2)
(0,0)	((0,1),(0,1))	((0,2),(0,2))	((0,0),(0,2))	((1,0),(1,0))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))
(0,1)	((0,1),(0,1))	((0,1),(0,2))	((0,0),(0,0))	((1,0),(1,0))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))
(0,2)	((0,2),(0,2))	((*,*),(*,*))	((0,2),(0,2))	((1,0),(1,0))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))
(1,0)	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))
(1,1)	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))
(1,2)	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))
(2,0)	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))
(2,1)	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))
(2,2)	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))	((*,*),(*,*))

FIG. 10

STATE \ INPUT	(0,0)	(0,1)	(0,2)	(1,0)
(0,0)	((0,1),(0,1))	((0,2),(0,2))	((0,0),(0,2))	((1,0),(1,0))
(0,1)	((0,1),(0,1))	((0,1),(0,2))	((0,0),(0,0))	((1,0),(1,0))
(0,2)	((0,2),(0,2))	———	((0,2),(0,2))	((1,0),(1,0))
(1,0)	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
(1,1)				

FIG. 11A

STATE \ INPUT	(0,0)	(0,1)	(0,2)	(1,0)
(0,0)	((0,1),(0,1))	((0,2),(0,2))	((0,0),(0,2))	((1,0),(1,0))
(0,1)	((0,1),(0,1))	((0,1),(0,2))	((0,0),(0,0))	((1,0),(1,0))
(0,2)	((0,2),(0,2))	———	((0,2),(0,2))	((1,0),(1,0))
(1,0)	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
(1,1)	((0,1),(0,1))	((0,1),(0,2))	((0,0),(0,0))	((1,0),(1,0))

FIG. 11B

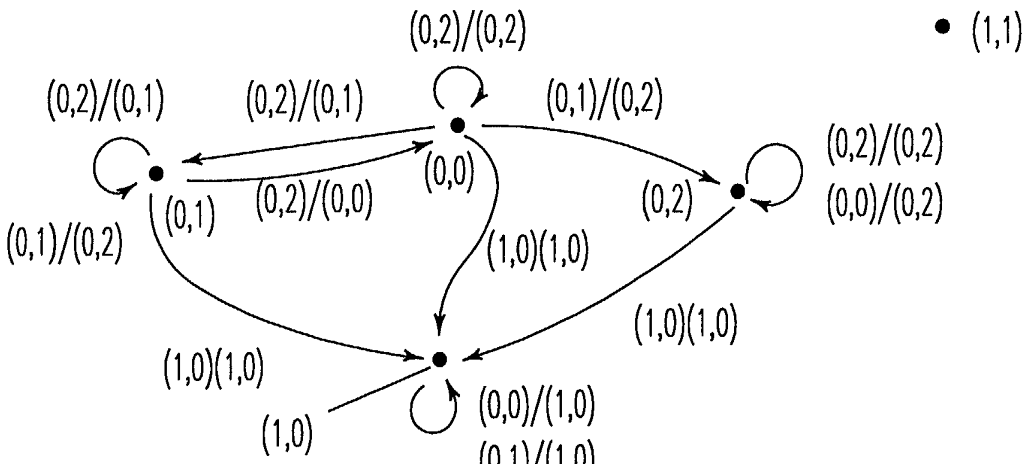


FIG. 11C

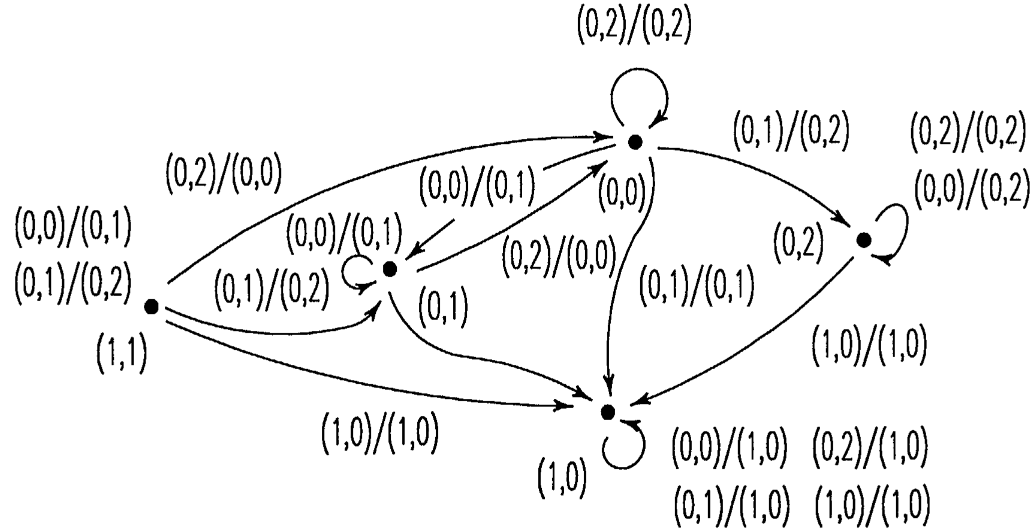


FIG. 11D

STATE \ INPUT	(0,0)	(0,1)	(0,2)	(1,0)
(0,0)	((0,1),(0,1))	((0,2),(0,2))	((0,0),(0,2))	((1,0),(1,0))
(0,1)	((0,1),(0,1))	((0,1),(0,2))	((0,0),(0,0))	((1,0),(1,0))
(0,2)	((0,2),(0,2))	———	((0,2),(0,2))	((1,0),(1,0))
(1,0)	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
(1,1)	((0,1),(0,1))	((0,1),(0,2))	((0,0),(0,0))	((1,0),(1,0))

FIG. 12A

STATE \ INPUT	(0,0)	(0,1)	(0,2)	(1,0)
(0,0)	((0,1),(0,1))	((0,2),(0,2))	((0,0),(0,2))	((1,0),(1,0))
(0,1)	((0,1),(0,1))	((1,1),(0,2))	((0,0),(0,0))	((1,0),(1,0))
(0,2)	((0,2),(0,2))	———	((0,2),(0,2))	((1,0),(1,0))
(1,0)	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
(1,1)	((0,1),(0,1))	((0,1),(0,2))	((0,0),(0,0))	((1,0),(1,0))

FIG. 12B

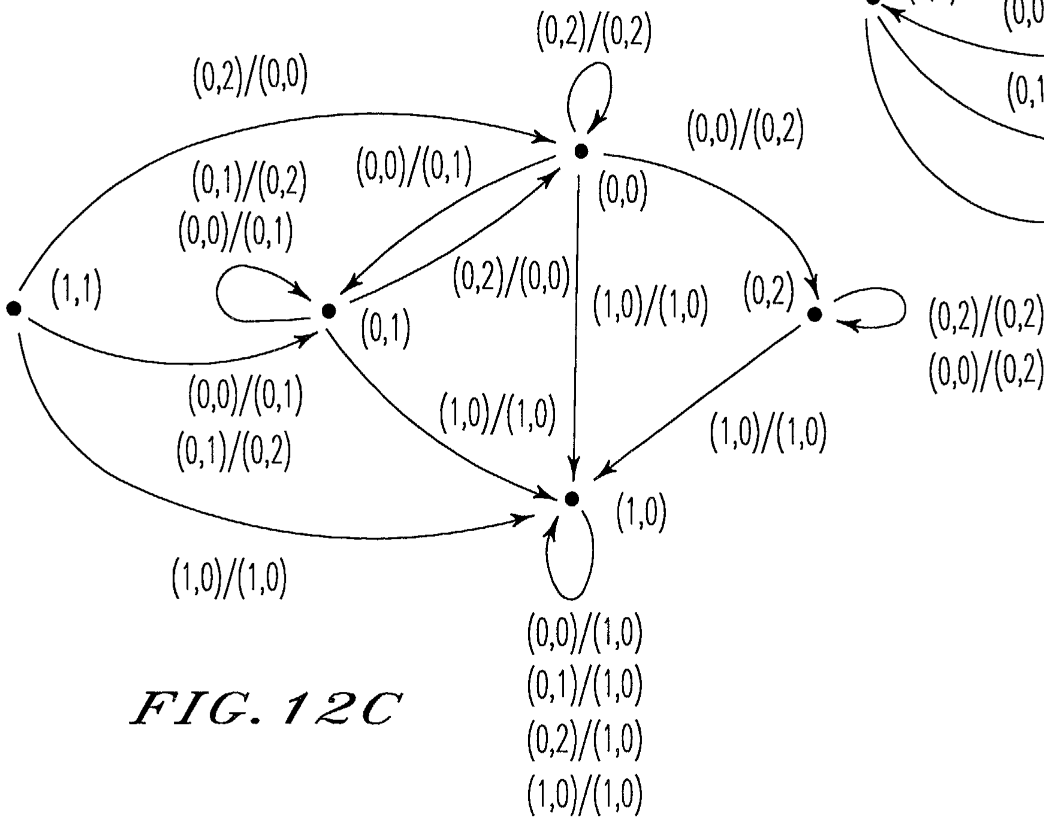


FIG. 12C

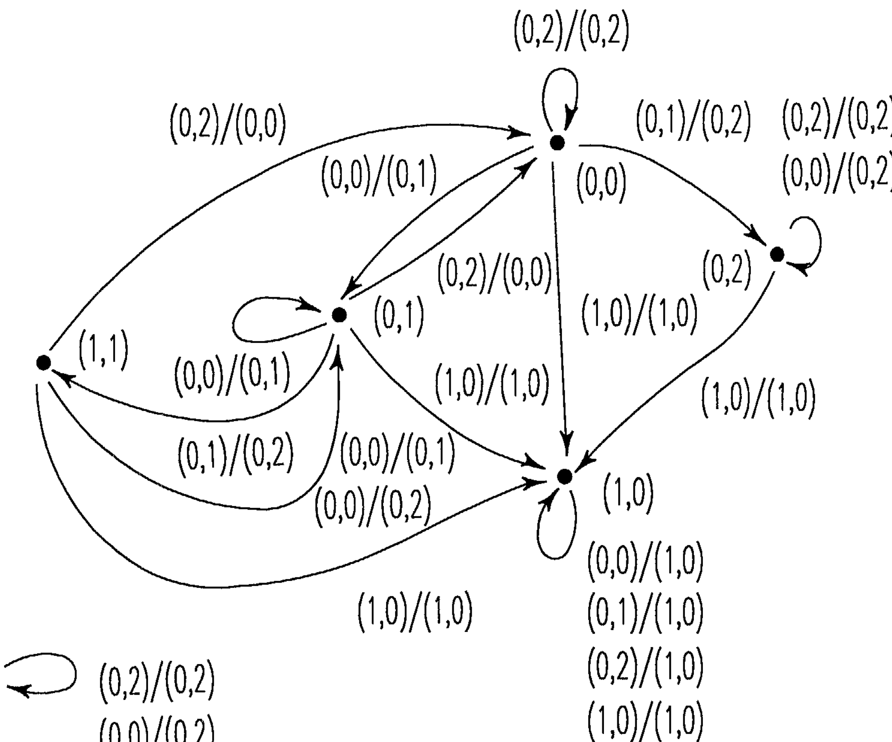


FIG. 12D

STATE \ INPUT	(0,0)	(0,1)	(0,2)	(1,0)
(0,0)	((0,1),(0,1))	((0,2),(0,2))	((0,0),(0,2))	((1,0),(1,0))
(0,1)	((0,1),(0,1))	((0,1),(0,2))	((0,0),(0,0))	((1,0),(1,0))
(0,2)	((0,2),(0,2))	———	((0,2),(0,2))	((1,0),(1,0))
(1,0)	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))	((1,0),(1,0))
(1,1)	((0,1),(0,1))	((0,1),(0,2))	((0,0),(0,0))	((1,0),(1,0))

FIG. 13A

STATE \ INPUT	(0,0)	(0,1)	(0,2)	(1,0)
(0,0)	((1,0),(0,1))	((0,2),(0,2))	((0,0),(0,2))	((0,1),(1,0))
(0,1)	((0,1),(1,0))	((0,1),(1,0))	((0,1),(1,0))	((0,1),(1,0))
(0,2)	((0,2),(0,2))	———	((0,2),(0,2))	((0,1),(1,0))
(1,0)	((1,0),(0,1))	((1,0),(0,2))	((0,0),(0,0))	((0,1),(1,0))
(1,1)	((1,0),(0,1))	((1,0),(0,2))	((0,0),(0,0))	((0,1),(1,0))

FIG. 13B

STATE \ INPUT	(0,0)	(0,1)	(0,2)	(1,0)
(0,0)	((1,0),(0,1))	((0,2),(0,2))	((0,0),(0,2))	((0,1),(1,0))
(0,1)	((0,1),(1,0))	((0,1),(1,0))	((0,1),(1,0))	((0,1),(1,0))
(0,2)	((0,2),(0,2))	—	((0,2),(0,2))	((0,1),(1,0))
(1,0)	((1,0),(0,1))	((1,0),(0,2))	((0,0),(0,0))	((0,1),(1,0))
(1,1)	((1,0),(0,1))	((1,0),(0,2))	((0,0),(0,0))	((0,1),(1,0))

FIG. 14A

STATE \ INPUT	(0,0)	(0,1)	(0,2)	(1,0)
(0,0)	((1,0),(0,1))	((0,2),(0,2))	((0,1),(1,0))	((0,0),(0,2))
(0,1)	((0,1),(1,0))	((0,1),(1,0))	((0,1),(1,0))	((0,1),(1,0))
(0,2)	((0,2),(0,2))	—	((0,1),(1,0))	((0,2),(0,2))
(1,0)	((1,0),(0,1))	((1,0),(0,2))	((0,1),(1,0))	((0,0),(0,0))
(1,1)	((1,0),(0,1))	((1,0),(0,2))	((0,1),(1,0))	((0,0),(0,0))

FIG. 14B

INPUT / STATE	(0,0)	(0,1)	(0,2)	(1,0)
(0,0)	((1,0),(0,1))	((0,2),(0,2))	((0,1),(1,0))	((0,0),(0,2))
(0,1)	((0,1),(1,0))	((0,1),(1,0))	((0,1),(1,0))	((0,1),(1,0))
(0,2)	((0,2),(0,2))	—	((0,1),(1,0))	((0,2),(0,2))
(1,0)	((1,0),(0,1))	((1,0),(0,2))	((0,1),(1,0))	((0,0),(0,0))
(1,1)	((1,0),(0,1))	((1,0),(0,2))	((0,1),(1,0))	((0,0),(0,0))

FIG. 15A

INPUT / STATE	(0,0)	(0,1)	(0,2)	(1,0)
(0,0)	((1,0),(0,1))	((0,2),(1,0))	((0,1),(0,2))	((0,0),(1,0))
(0,1)	((0,1),(0,2))	((0,1),(0,2))	((0,1),(0,2))	((0,1),(0,2))
(0,2)	((0,2),(1,0))	—	((0,1),(0,2))	((0,2),(1,0))
(1,0)	((1,0),(0,1))	((1,0),(1,0))	((0,1),(0,2))	((0,0),(0,0))
(1,1)	((1,0),(0,1))	((1,0),(1,0))	((0,1),(0,2))	((0,0),(0,0))

FIG. 15B

STATE \ INPUT	(0,0)	
(0,0)	((0,1)(0,1))	

FIG. 16A

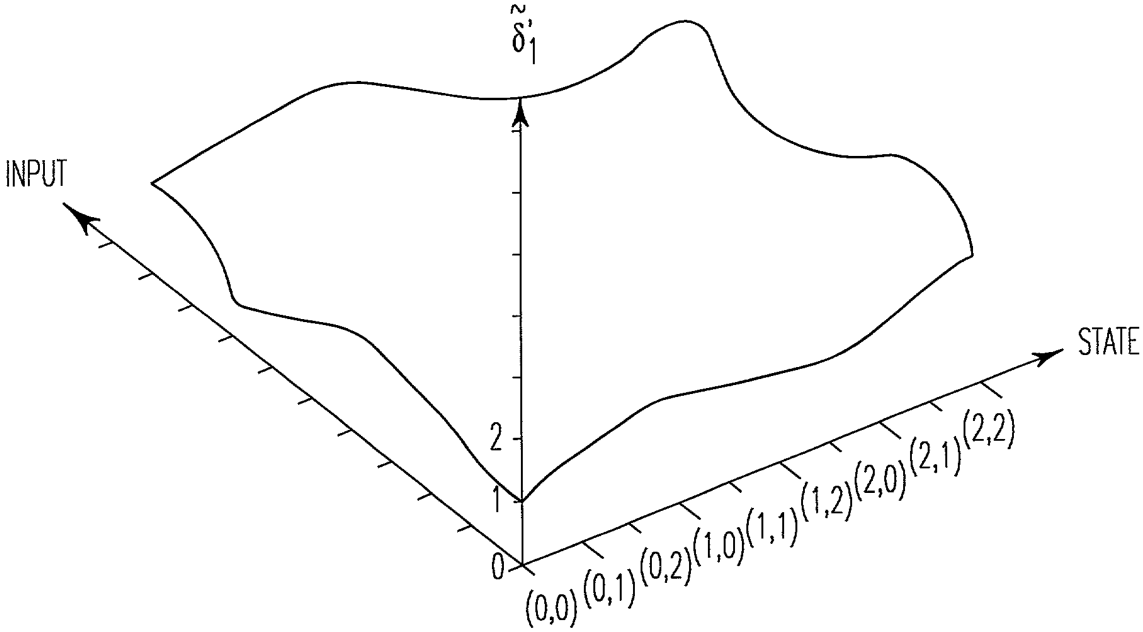


FIG. 16B

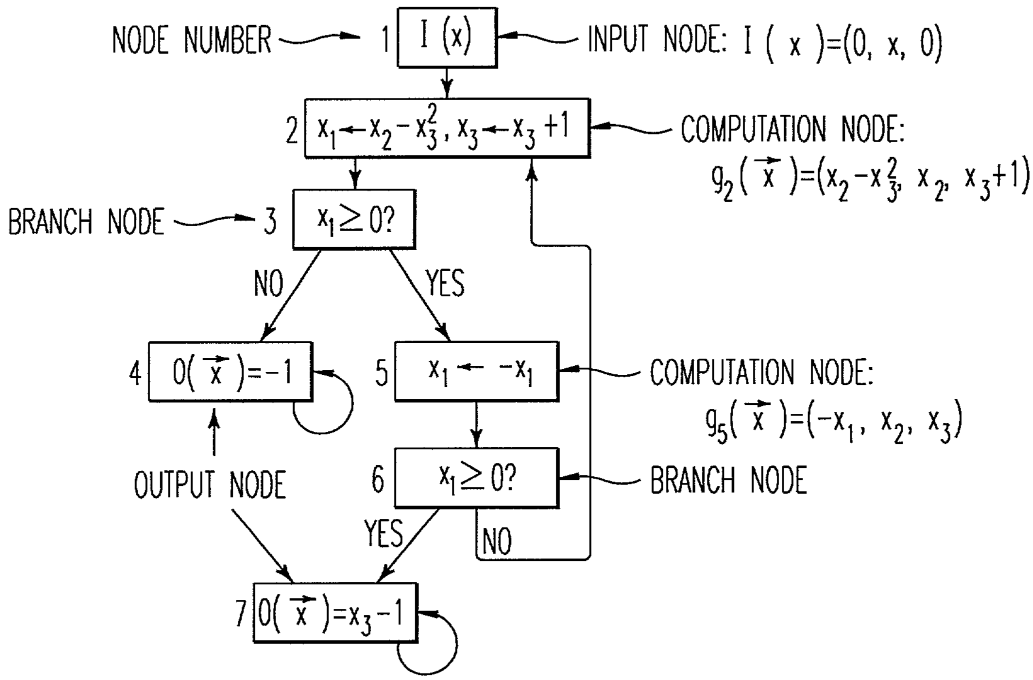
PRECALCULATE $a_i(x)$ FOR $k=\{0,1,2,4,5,\}<\mathbb{Z}_{11}$.

PRECOMPUTATION RESULTS IN THE SERIES OF POLYNOMIALS

$a_0(x)$
$a_1(x)$
$a_2(x)$
$a_4(x)$
$a_5(x)$

REPRESENTED BY THEIR RESPECTIVE ARRAYS OF COEFFECIENTS

FIG. 17



- WHEN RESTRICTING A BSS MACHINE TO A FINITE FIELD $\mathbb{Z}_N$, THE CHOICE OF N IS DICTATED BY THE FOLLOWING:
  1) N MUST BE A PRIME NUMBER
  2) N MUST BE AT LEAST AS GREAT AS THE NUMBER OF NODES
  3) N MUST MAKE ALLOWANCE FOR CONSTANTS USED IN THE MACHINE
  4) N MUST ACCOMODATE USER REQUIREMENTS
- FOR THE ABOVE EXAMPLE:
  N SATISFIES THE FIRST CONDITION IF IT IS EQUAL TO 2, 3, 5, 7, 11,...
  N SATISFIES THE SECOND CONDITION IF IT IS ≥ 7
  N THE GREATEST CONSTANTS HAVE ABSOLUTE VALUE 1, SO N SATISFIES THE THIRD CONDITION IF IT IS ≥ 2
  IF THE USER REQUIRES THAT THE x INPUT MUST BE ABLE TO BE AS LARGE AS 100, N SATISFIES THE FOURTH CONDITION IF IT IS > 100. THE LEAST N SATISFYING ALL FOUR CONDITIONS WOULD THEN BE N=101
- SINCE ALL MAPPINGS IN THE BSS MACHINE ABOVE ARE POLYNOMIAL, THE RESTRICTION OF COMPUTATION MAPPINGS TO POLYNOMIAL MAPPINGS IS ALREADY SATISFIED.
- THE NEW NODE-NUMBERING CONVENTION SIMPLY SUBTRACTS 1 FROM EACH NODE NUMBER, SUCH THAT NUMBERING BEGINS AT 0.

1	2	3	4	5	6	7
↓	↓	↓	↓	↓	↓	↓
0	1	2	3	4	5	6

FIG. 18

THE FULL STATE SPACE OF THE BSS MACHINE, AS ADAPTED SO FAR, IS:

$$\underbrace{\{0,\ldots,6\}}_{\text{NODE NUMBER SPACE}} \times \underbrace{\mathbb{Z}_N \times \mathbb{Z}_N \times \mathbb{Z}_N}_{\text{STATE SPACE}}$$

CORRESPONDING VECTORS HAVE THE COMPONENTS:

n	x_1	x_2	x_3

THE REVISED FULL STATE SPACE ADDS THE OUTPUT AND INPUT COMPONENTS:

$$\{0,\ldots,6\} \times \mathbb{Z}_N \times \mathbb{Z}_N \times \mathbb{Z}_N \times \underbrace{\mathbb{Z}_N}_{\text{OUTPUT}} \times \underbrace{\mathbb{Z}_N}_{\text{INPUT}}$$

CORRESPONDING VECTORS HAVE THE COMPONENTS:

n	x_1	x_2	x_3	x_4	x_5
				OUTPUT	INPUT

ALSO A COMPUTATION MAPPING g_i IS ADDED TO EVERY NODE RHAT DOESN'T ALREADY HAVE ONE. THUS FOR EACH NODE VIEWED IN ISOLATION:

NODE 0: $g_0(\vec{x})=(0, x_5, 0, 0, x_5)$ IS ADDED

NODE 1: "g_2"(NOW g_1) IS CHANGED TO $g_1(\vec{x})=(x_2 - x_3^2, x_2, x_3+1, 0, x_5)$

NODE 2: $g_2(\vec{x})=(x_1, x_2, x_3, 0, x_5)$ IS ADDED

NODE 3: $g_3(\vec{x})=(x_1, x_2, x_3^{N-1}, x_5)$ IS ADDED

NODE 4: g_4 (PREVIOUSLY "g_5") IS CHANGED TO $g_4(\vec{x})=(-x_1, x_2, x_3, 0, x_5)$

NODE 5: $g_5(\vec{x})=(x_1, x_2, x_3, 0, x_5)$ IS ADDED

NODE 6: $g_6(\vec{x})=(x_1, x_2, x_3, x_3-1, x_5)$ IS ADDED

AS THE RELATION ≥ 0 HOLDS FOR ALL ELEMENTS IN $\mathbb{Z}_N$, IT IS REPLACED BY A SERIES OF SET INCLUSION RELATIONS. BECAUSE $\mathbb{Z}_N$ DOES NOT HAVE NEGATIVE NUMBERS AS ELEMENTS, THE RELATIONS WILL NOT HAVE AN EXACT CORRESPONDENCE TO THE ORIGINAL RELATIONS. REASONABLE SET INCLUSION RELATIONS FOR THIS EXAMPLE ARE:

FOR NODE 2: $\in Zp-\{0\}$ WITH THE SAME MAPPING IN NODE 1 AS BEFORE.
FOR NODE 5: $\in \{1\}$, CHANGING g_4 TO $g_4(\vec{x})=(x_3+1, x_2, x_3, 0, x_5)$

FIG. 19

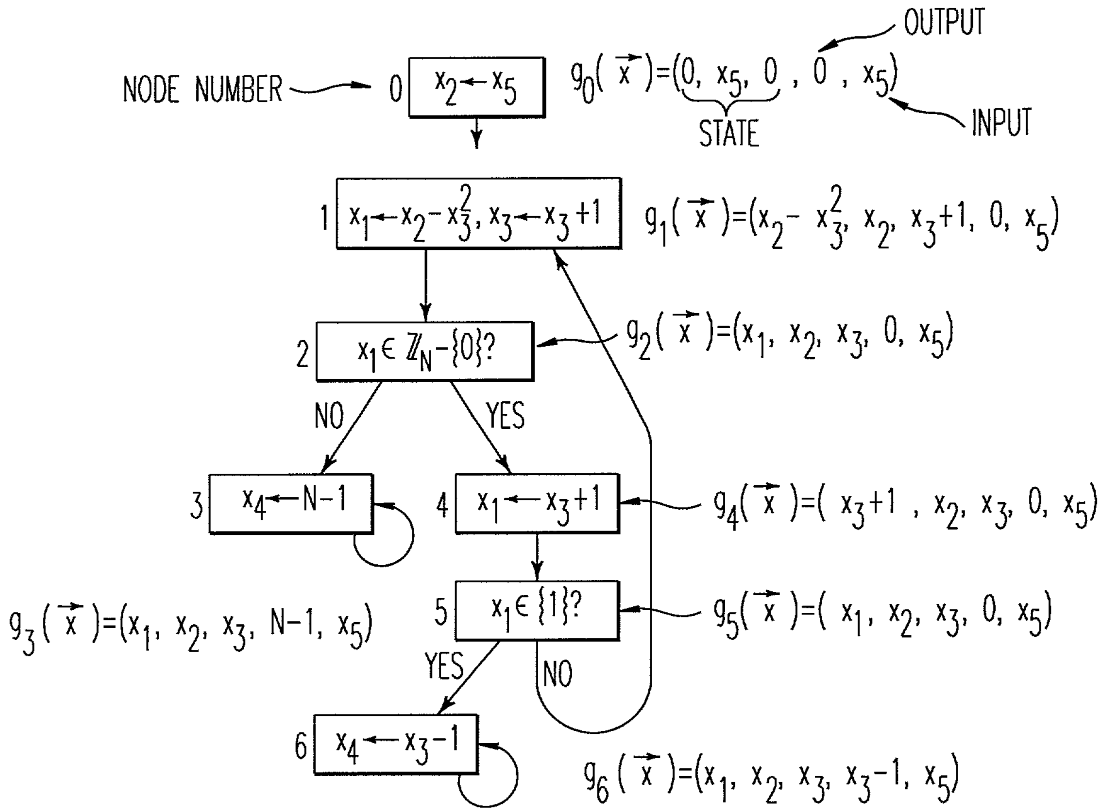


FIG. 20A

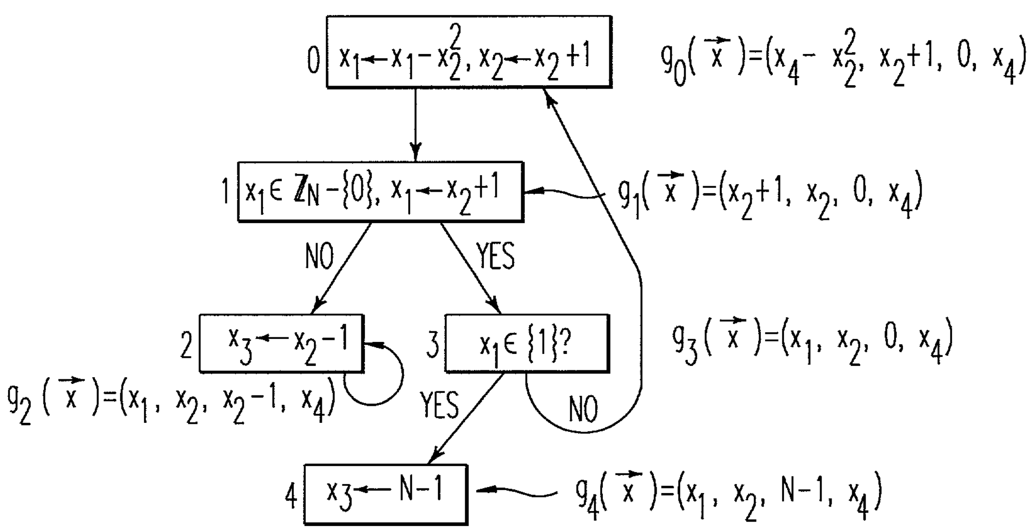
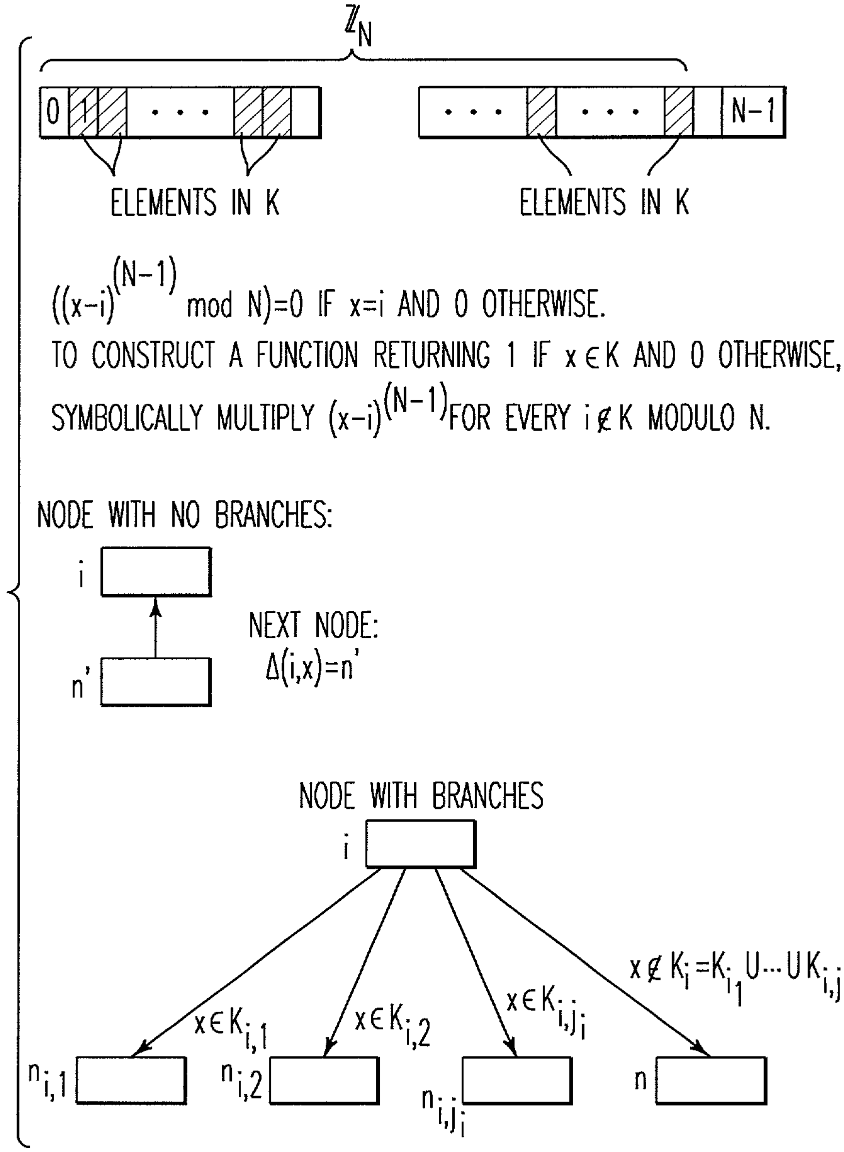


FIG. 20B



$\mathbb{Z}_N$
0
1
N−1
ELEMENTS IN K
ELEMENTS IN K
$((x-i)^{(N-1)} \bmod N)=0$ IF $x=i$ AND 0 OTHERWISE.
TO CONSTRUCT A FUNCTION RETURNING 1 IF $x \in K$ AND 0 OTHERWISE,
SYMBOLICALLY MULTIPLY $(x-i)^{(N-1)}$ FOR EVERY $i \notin K$ MODULO N.
NODE WITH NO BRANCHES:
i
n'
NEXT NODE:
$\Delta(i,x)=n'$
NODE WITH BRANCHES
i
$x \in K_{i,1}$
$x \in K_{i,2}$
$x \in K_{i,j_i}$
$x \notin K_i = K_{i_1} \cup \dots \cup K_{i,j_i}$
$n_{i,1}$
$n_{i,2}$
n_{i,j_i}
n

FIG. 21

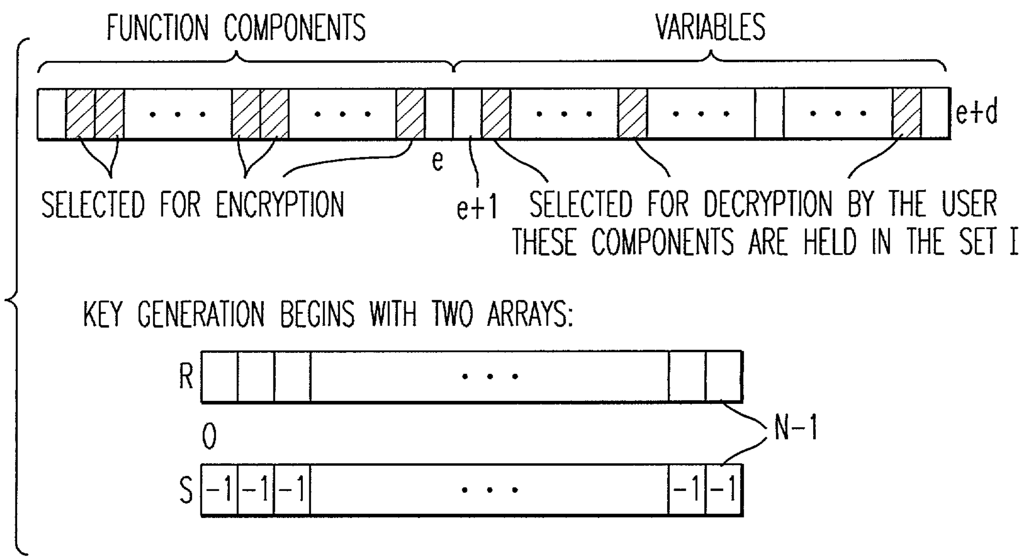


FIG. 22A

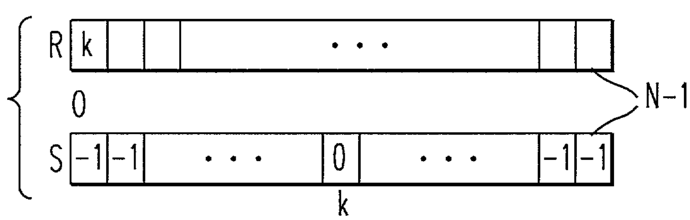


FIG. 22B

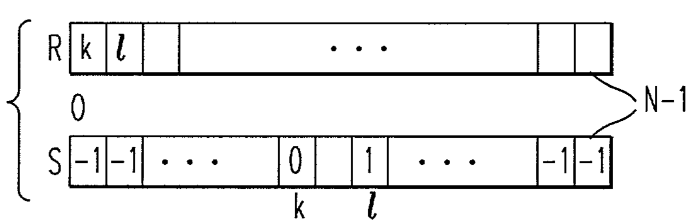


FIG. 22C

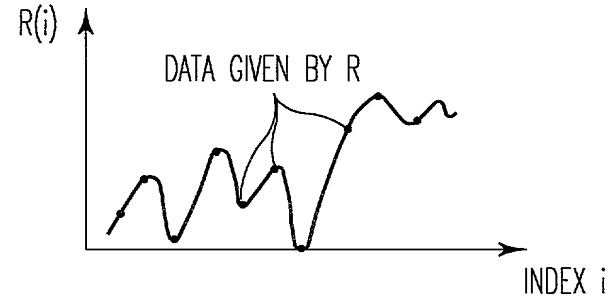


FIG. 23

X Y MOD 5

X \ Y	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

FIG. 24A

X^Y MOD 5

X \ Y	0	1	2	3	4
0	1	0	0	0	0
1	1	1	1	1	1
2	1	2	4	3	1
3	1	3	4	2	1
4	1	4	1	4	1

FIG. 24B

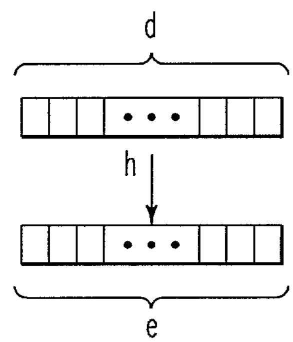


FIG. 25A

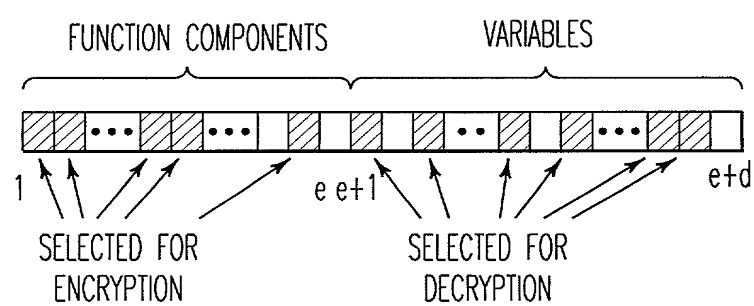


FIG. 25B

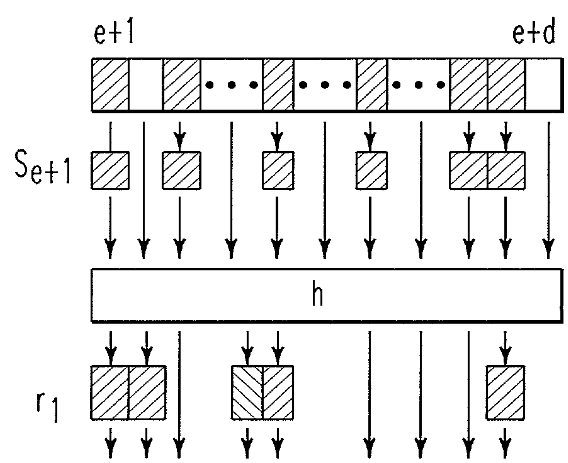


FIG. 25C

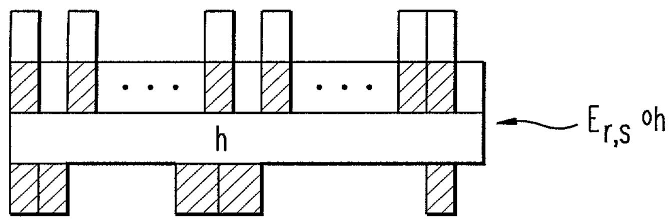


FIG. 26A

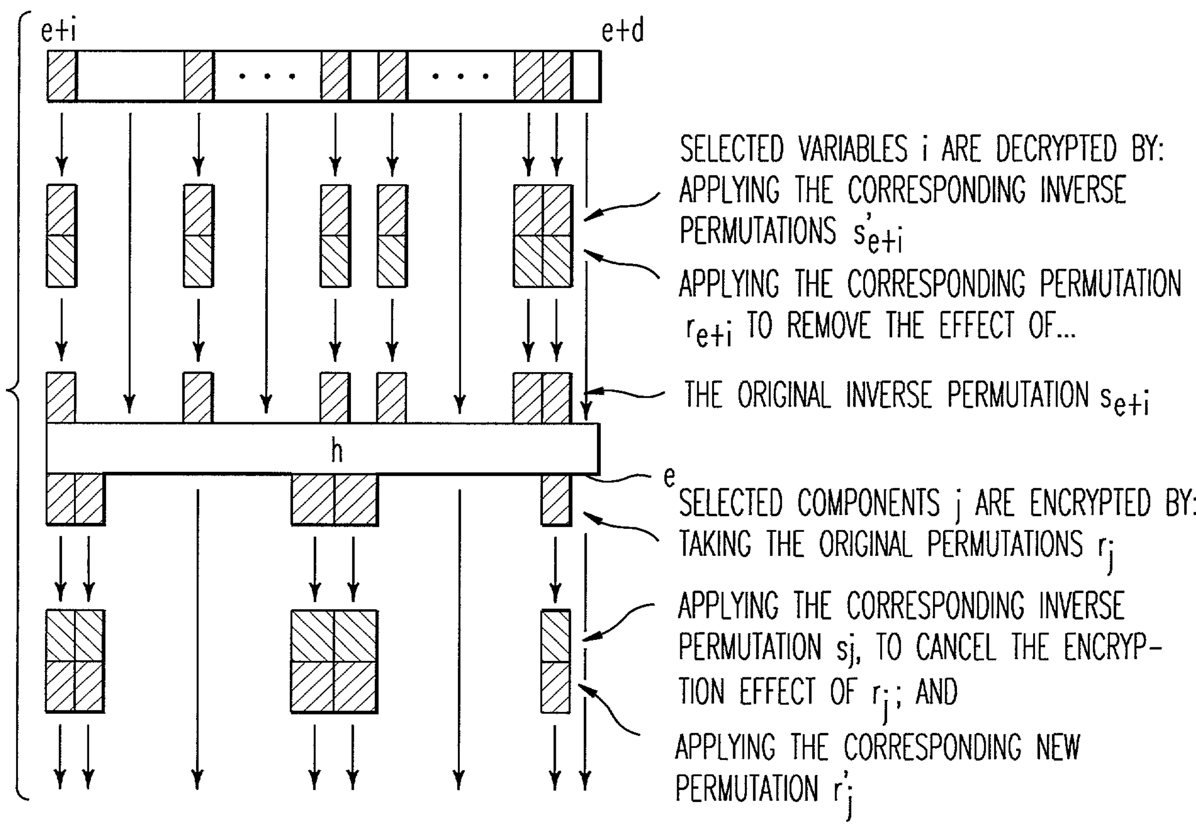


FIG. 26B

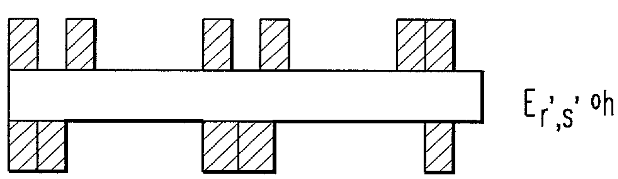


FIG. 26C

X_1 \ X_2	0	1	2	3	4
0	(3,4)	(1,2)	(4,0)	(2,1)	(1,3)
1	(0,0)	(2,3)	(3,4)	(4,1)	(0,2)
2	(2,0)	(3,2)	(1,2)	(0,1)	(1,4)
3	(4,0)	(2,0)	(4,4)	(4,4)	(2,4)
4	(1,1)	(2,2)	(1,0)	(4,1)	(4,2)

FIG. 27A

X	0	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
t_f	23	2	4	6	11	17	13	2	12	4	23	11	24	1	7	9	5	24	9	16	10	21	22	14

FIG. 27B

FUNCTION TABLE FOR f

1	2	3	4	5

FIG. 27C

FUNCTION TABLE FOR t_f

1	2	3	4	5

FIG. 27D

X1 \ X2	0	1	2	3	4
0	0	1	3	2	0
1	2	2	4	2	0
2	1	0	4	2	1
3	2	3	3	2	1
4	2	0	1	1	2

FIG. 28A

X	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
tf	23	0	2	4	6	11	17	13	2	12	4	23	11	24	1	7	9	5	24	9	16	10	21	22	14

FIG. 28B

X	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	13	16	17	18	19	20	21	22	23	24
tg	0	2	1	2	2	1	2	0	3	0	3	4	4	3	1	2	2	2	2	1	0	0	1	1	2

FIG. 28C

X	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
tgf	1	0	1	2	2	4	2	3	1	4	2	1	4	2	2	0	0	1	2	0	2	3	0	1	1

FIG. 28D

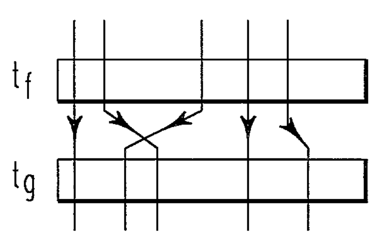


FIG. 28E

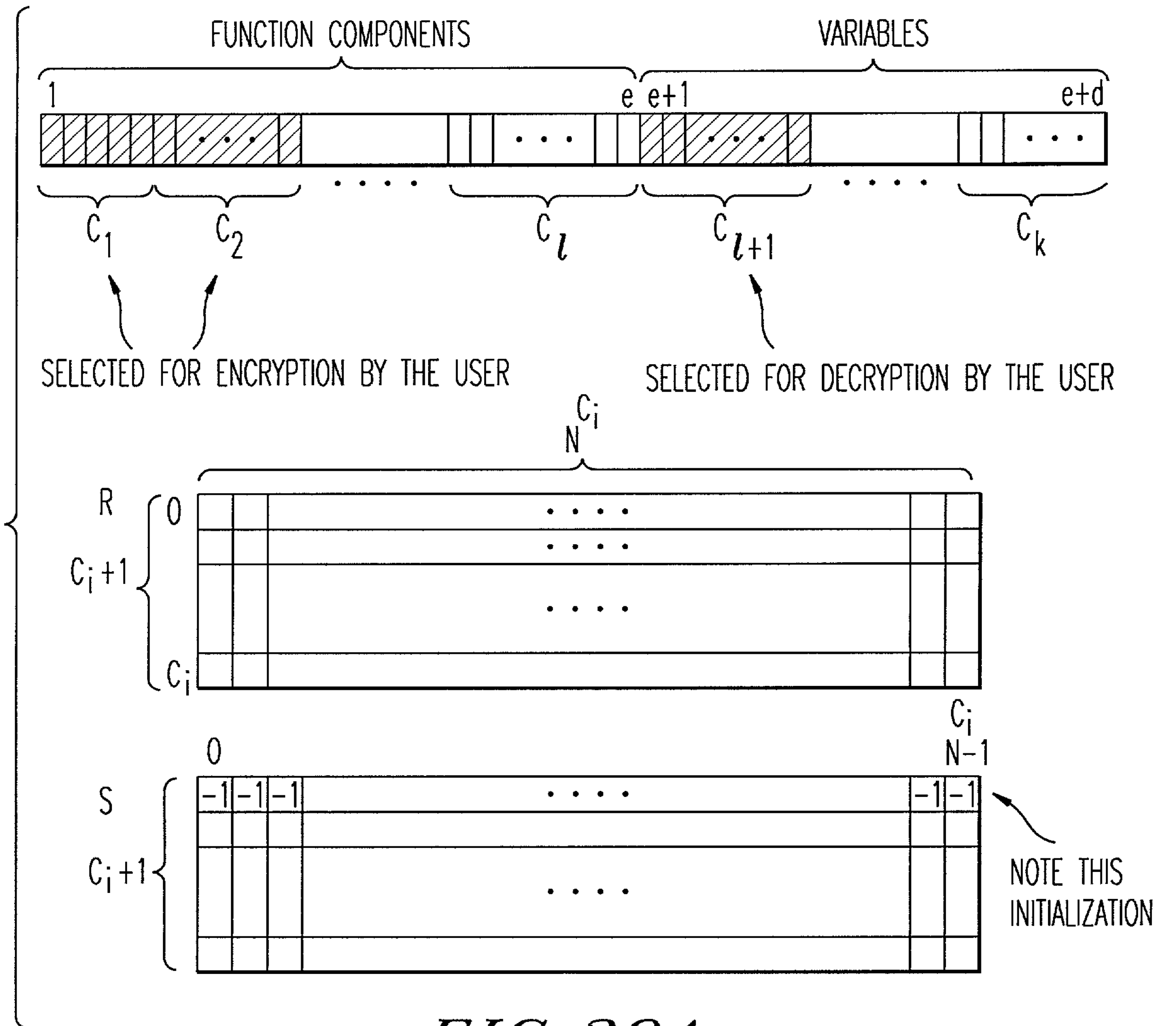


FIG. 29A

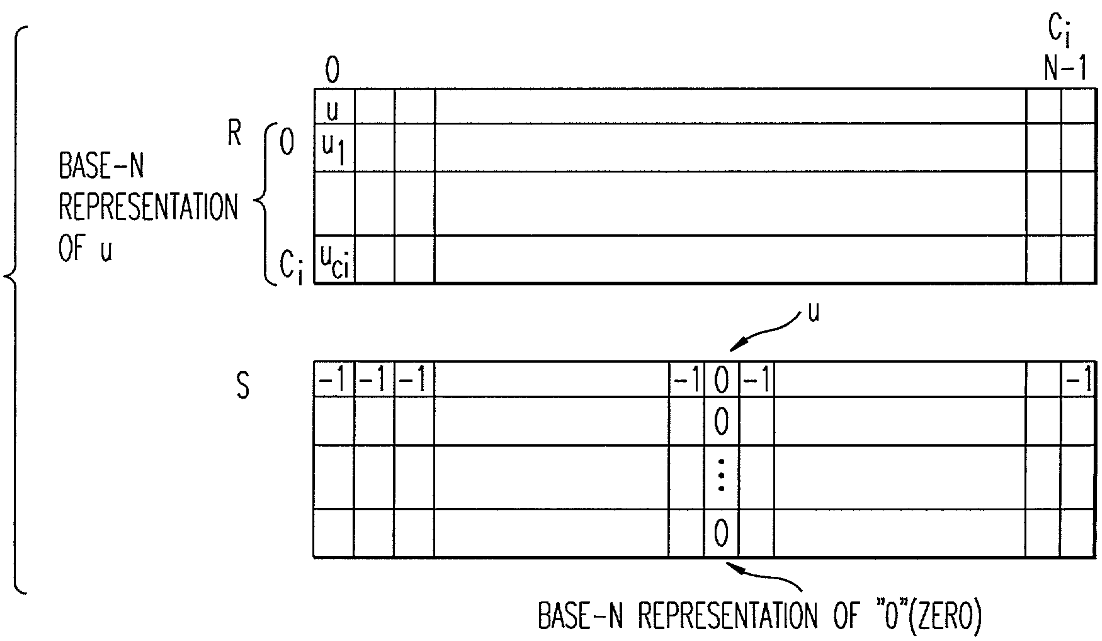


FIG. 29B

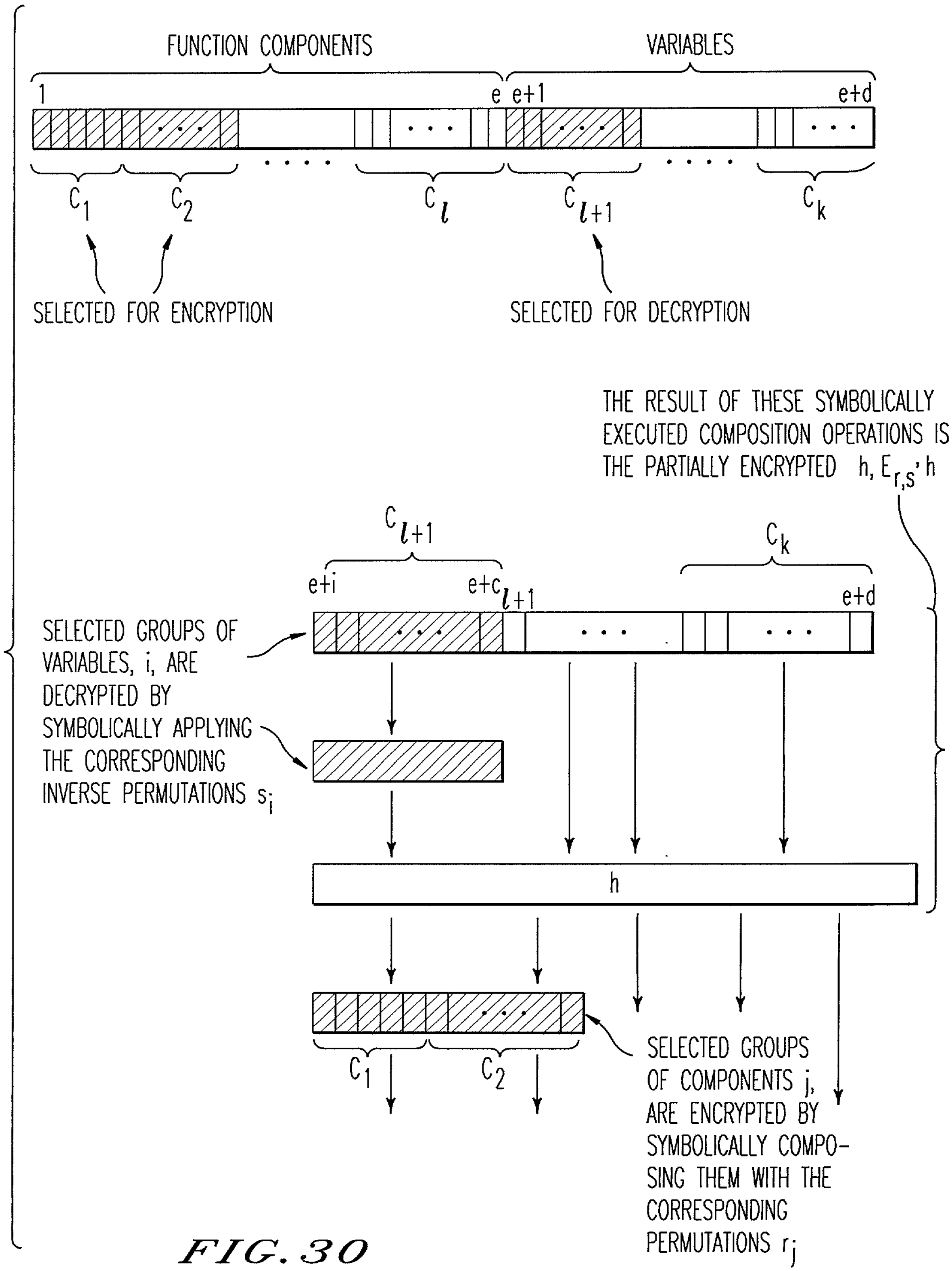

FUNCTION COMPONENTS
VARIABLES
1
e
e+1
e+d
C1
C2
Cl
Cl+1
Ck
SELECTED FOR ENCRYPTION
SELECTED FOR DECRYPTION
THE RESULT OF THESE SYMBOLICALLY EXECUTED COMPOSITION OPERATIONS IS THE PARTIALLY ENCRYPTED h, Er,s' h
Cl+1
Ck
e+i
e+cl+1
e+d
SELECTED GROUPS OF VARIABLES, i, ARE DECRYPTED BY SYMBOLICALLY APPLYING THE CORRESPONDING INVERSE PERMUTATIONS si
h
C1
C2
SELECTED GROUPS OF COMPONENTS j, ARE ENCRYPTED BY SYMBOLICALLY COMPOSING THEM WITH THE CORRESPONDING PERMUTATIONS rj

FIG. 30

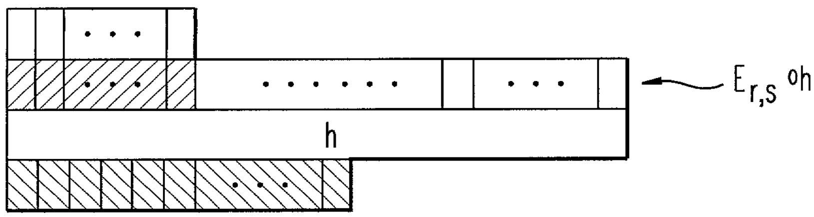


FIG. 31A

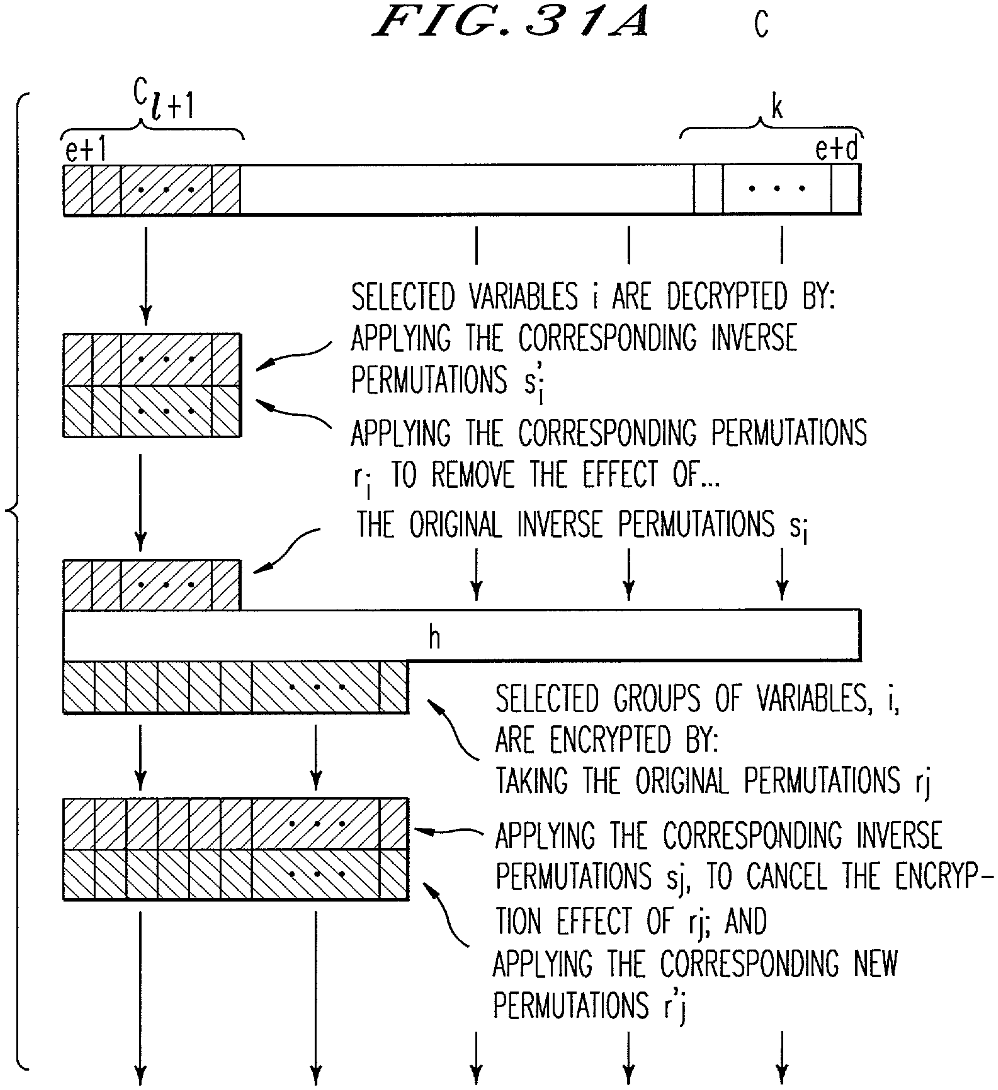


FIG. 31B

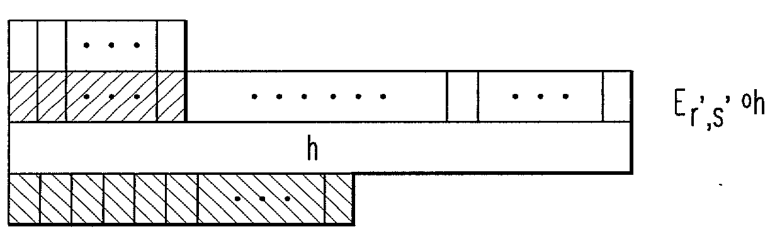


FIG. 31C

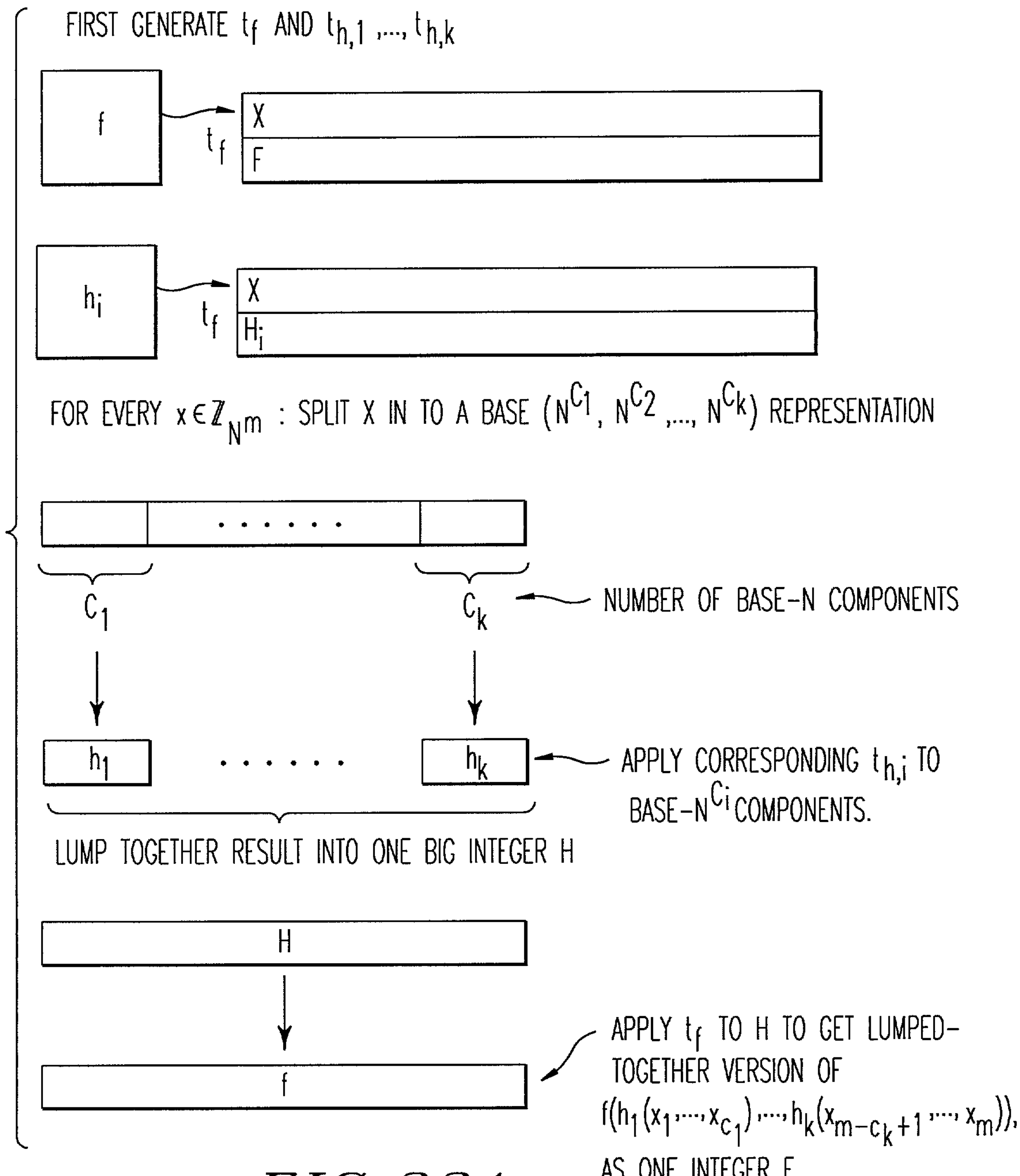


FIG. 32A

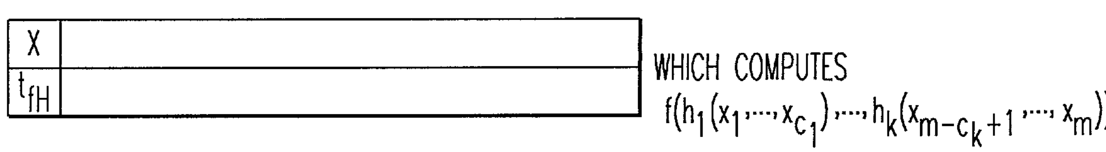


FIG. 32B

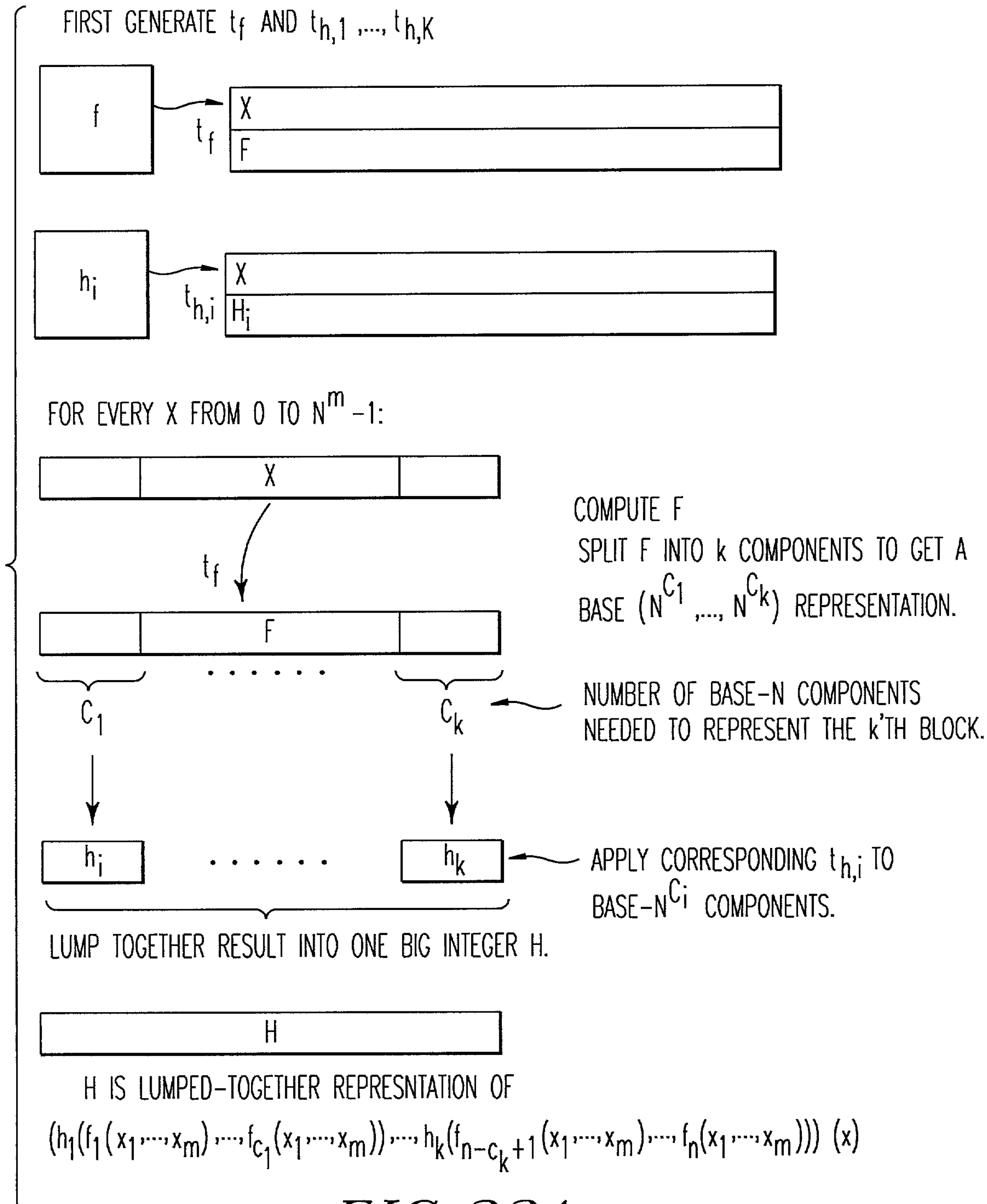


FIG. 33A

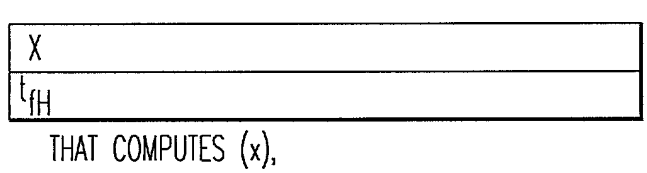


FIG. 33B

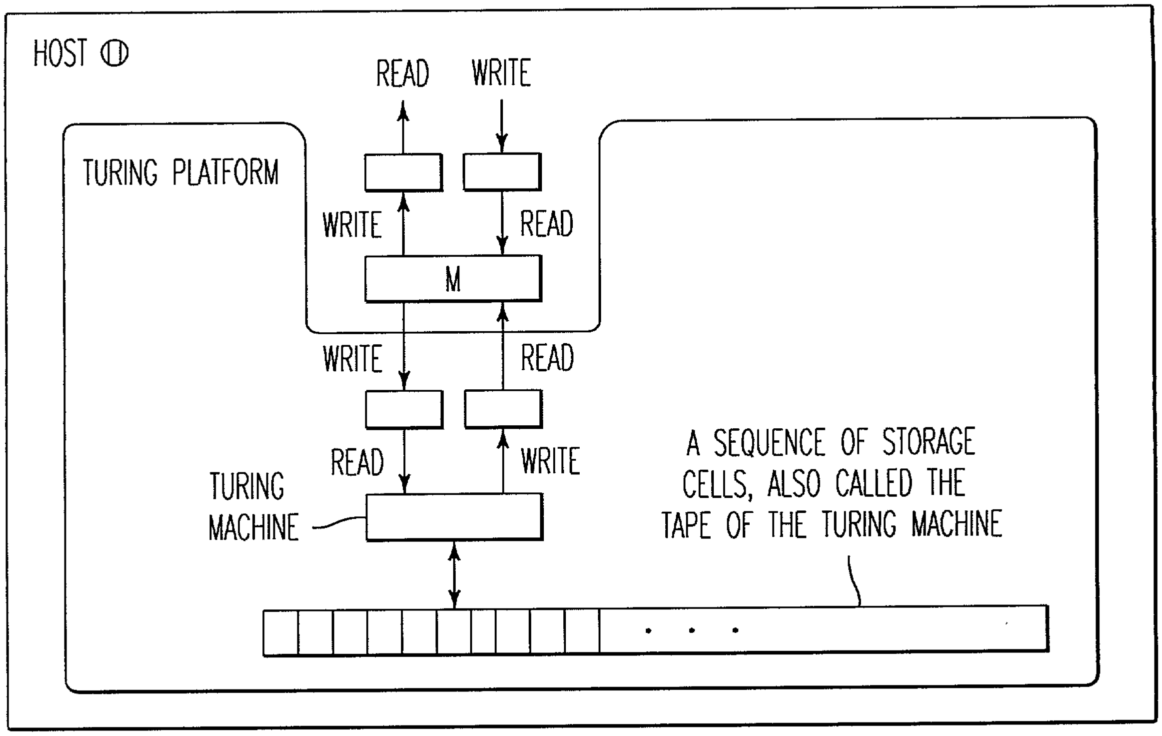

HOST ①
TURING PLATFORM
READ
WRITE
WRITE
READ
M
WRITE
READ
READ
WRITE
TURING
MACHINE
A SEQUENCE OF STORAGE
CELLS, ALSO CALLED THE
TAPE OF THE TURING MACHINE

FIG. 34

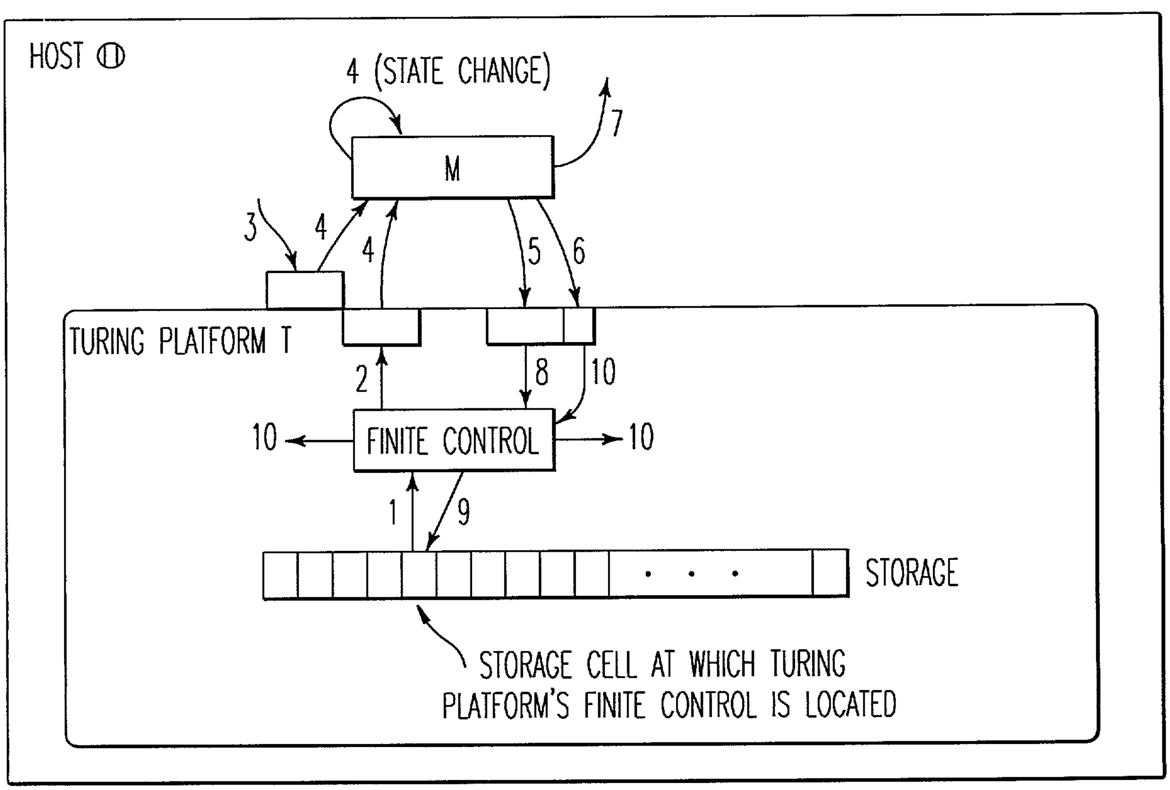

HOST ①
4 (STATE CHANGE)
M
7
3
4
4
5
6
TURING PLATFORM T
2
8
10
10
FINITE CONTROL
10
1
9
STORAGE
STORAGE CELL AT WHICH TURING
PLATFORM'S FINITE CONTROL IS LOCATED

FIG. 35

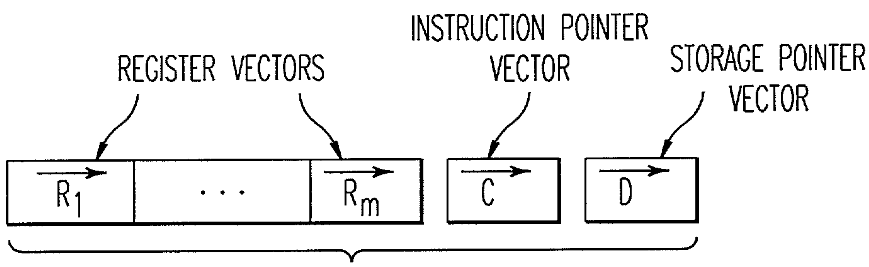


FIG. 36A

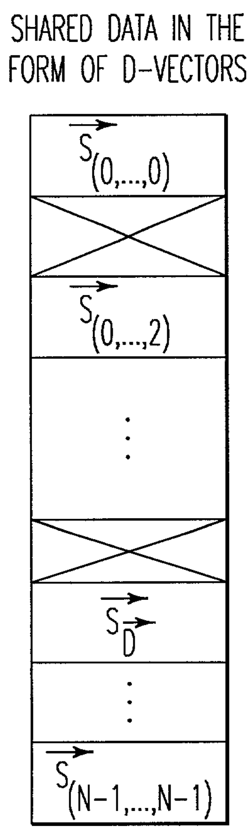


FIG. 36B

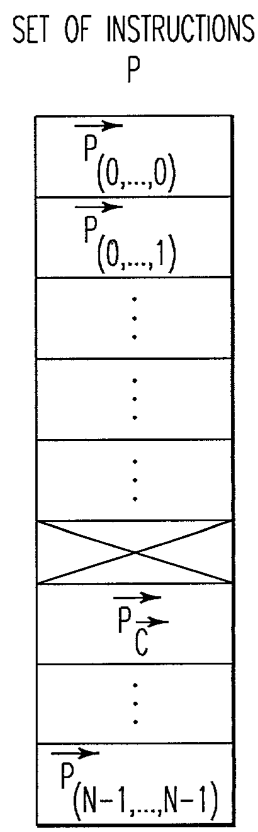


FIG. 36C

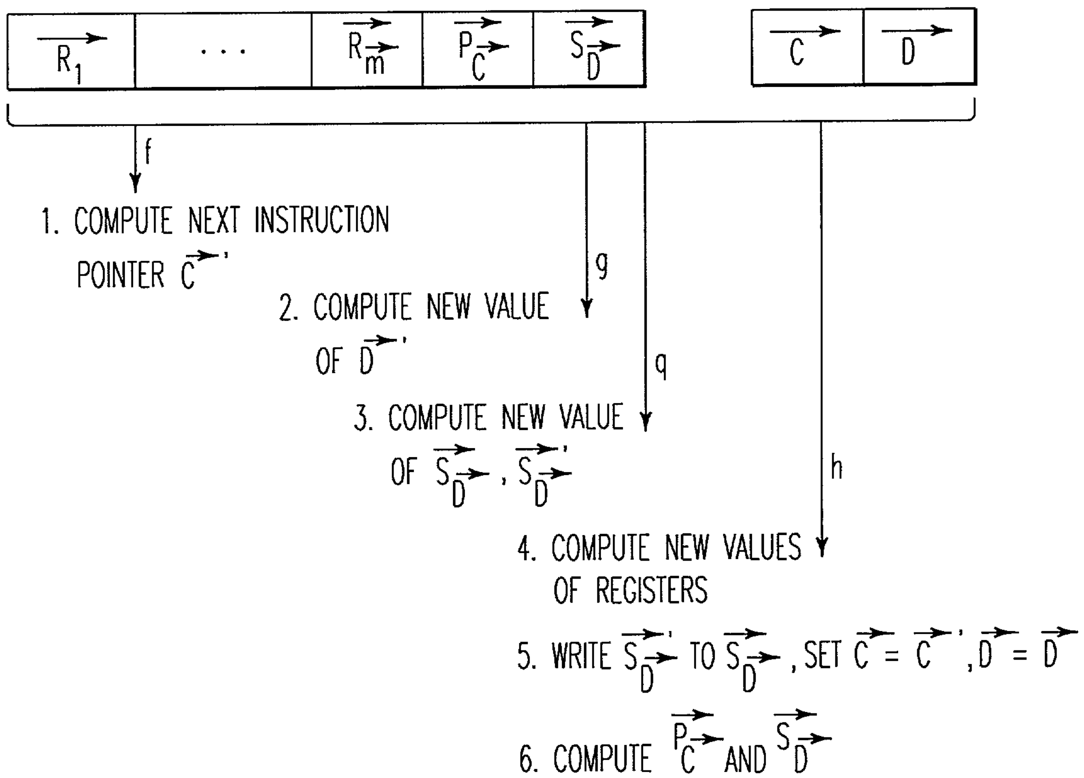


FIG. 36D

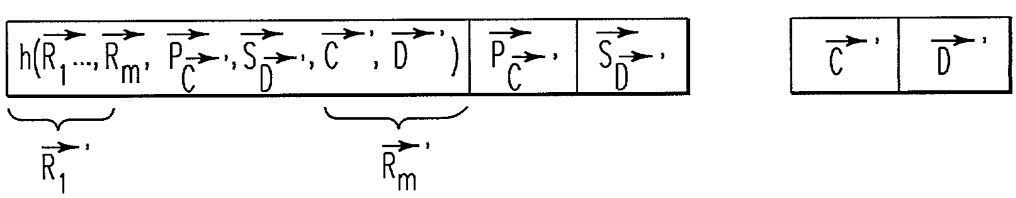


FIG. 36E

$h_1: \mathbb{Z}_2^2 \longrightarrow \mathbb{Z}_2^2$

$\vec{X}$	$h(\vec{X})$
(0,0)	(1,0)
(0,1)	(1,1)
(1,0)	(0,0)
(1,1)	(0,1)

2. COMPONENT

1. COMPONENT

$h_2: \mathbb{Z}_2^2 \longrightarrow \mathbb{Z}_2^2$

$\vec{X}$	$h(\vec{X})$
(0,0)	(1,1)
(0,1)	(0,0)
(1,0)	(1,0)
(1,1)	(1,0)

FIG. 37A

$f: \mathbb{Z}_2^4 \longrightarrow \mathbb{Z}_2^3$

$\vec{X}$	$f(\vec{X})$
(0,0,0,0)	(1,0,1)
(0,0,0,1)	(0,0,1)
(0,0,1,0)	(1,0,1)
(0,0,1,1)	(0,0,0)
(0,1,0,0)	(1,0,0)
(0,1,0,1)	(0,0,0)
(0,1,1,0)	(1,1,1)
(0,1,1,1)	(1,0,0)
(1,0,0,0)	(1,1,0)
(1,0,0,1)	(0,0,1)
(1,0,1,0)	(0,1,1)
(1,0,1,1)	(1,0,1)
(1,1,0,0)	(1,1,1)
(1,1,0,1)	(0,0,0)
(1,1,1,0)	(1,1,0)
(1,1,1,1)	(0,1,0)

FIG. 37B

$g: \mathbb{Z}_2^3 \longrightarrow \mathbb{Z}_2^3$

$\vec{X}$	$g(\vec{X})$
(0,0,0)	(1,0,1)
(0,0,1)	(0,0,0)
(0,1,0)	(1,1,1)
(0,1,1)	(1,0,0)
(1,0,0)	(0,1,1)
(1,0,1)	(1,0,1)
(1,1,0)	(1,1,0)
(1,1,1)	(1,1,1)

FIG. 37C

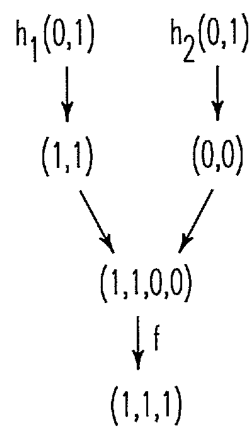


FIG. 37D

$f : \mathbb{Z}_2^4 \longrightarrow \mathbb{Z}_2^3$

$\vec{X}$	$f(\vec{X})$
(0,0,0,0)	(1,0,1)
(0,0,0,1)	(0,0,1)
(0,0,1,0)	(1,0,1)
(0,0,1,1)	(0,0,0)
(0,1,0,0)	(1,0,0)
(0,1,0,1)	(0,0,0)
(0,1,1,0)	(1,1,1)
(0,1,1,1)	(1,0,0)
(1,0,0,0)	(1,1,0)
(1,0,0,1)	(0,0,1)
(1,0,1,0)	(0,1,1)
(1,0,1,1)	(1,0,1)
(1,1,0,0)	(1,1,1)
(1,1,0,1)	(0,0,0)
(1,1,1,0)	(1,1,0)
(1,1,1,1)	(0,1,0)

FIG. 38A

$h_1 : \mathbb{Z}_2^2 \longrightarrow \mathbb{Z}_2^2$

$\vec{X}$	$h(\vec{X})$
(0,0)	(1,0)
(0,1)	(1,1)
(1,0)	(0,0)
(1,1)	(0,1)

$h_2 : \mathbb{Z}_2^2 \longrightarrow \mathbb{Z}_2^2$

$\vec{X}$	$h(\vec{X})$
(0,0)	(1,1)
(0,1)	(0,0)
(1,0)	(1,0)
(1,1)	(1,0)

FIG. 38B

$g : \mathbb{Z}_2^4 \longrightarrow \mathbb{Z}_2^4$

$\vec{X}$	$g(\vec{X})$
(0,0,0,0)	(0,0,0,0)
(0,0,0,1)	(1,1,0,0)
(0,0,1,0)	(0,1,0,0)
(0,0,1,1)	(1,0,1,1)
(0,1,0,0)	(0,0,1,0)
(0,1,0,1)	(1,0,1,1)
(0,1,1,0)	(0,1,1,0)
(0,1,1,1)	(0,0,1,1)
(1,0,0,0)	(0,0,1,0)
(1,0,0,1)	(1,1,0,0)
(1,0,1,0)	(1,1,1,0)
(1,0,1,1)	(0,1,0,0)
(1,1,0,0)	(0,1,1,0)
(1,1,0,1)	(1,0,1,1)
(1,1,1,0)	(0,0,1,0)
(1,1,1,1)	(1,0,1,0)

FIG. 38C

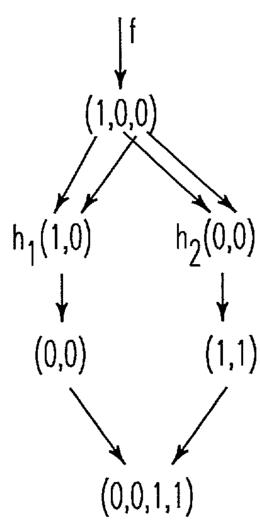
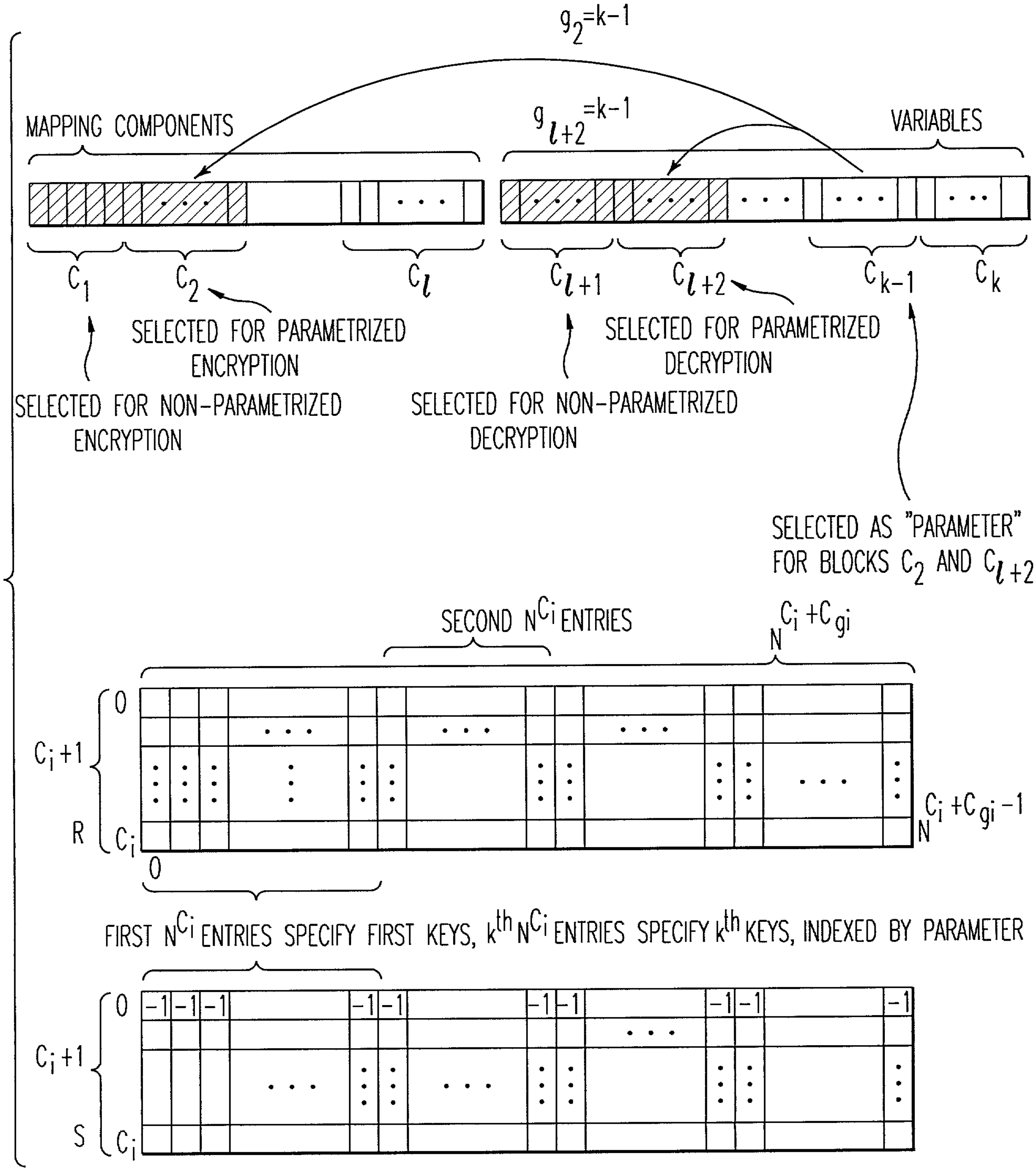


FIG. 38D



g_2=k-1
MAPPING COMPONENTS
g_{l+2}=k-1
VARIABLES
C_1
C_2
C_l
C_{l+1}
C_{l+2}
C_{k-1}
C_k
SELECTED FOR PARAMETRIZED ENCRYPTION
SELECTED FOR NON-PARAMETRIZED ENCRYPTION
SELECTED FOR PARAMETRIZED DECRYPTION
SELECTED FOR NON-PARAMETRIZED DECRYPTION
SELECTED AS "PARAMETER" FOR BLOCKS C_2 AND C_{l+2}
SECOND N^{C_i} ENTRIES
$N^{C_i+C_{gi}}$
C_i+1
R
0
C_i
$N^{C_i+C_{gi}-1}$
FIRST N^{C_i} ENTRIES SPECIFY FIRST KEYS, k^{th} N^{C_i} ENTRIES SPECIFY k^{th} KEYS, INDEXED BY PARAMETER
S
-1

FIG. 39A

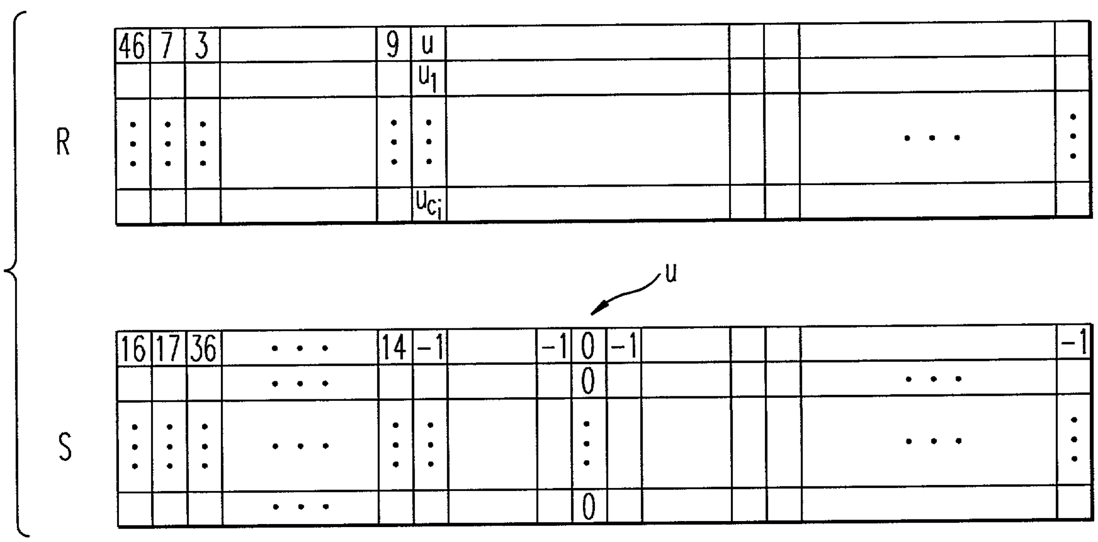
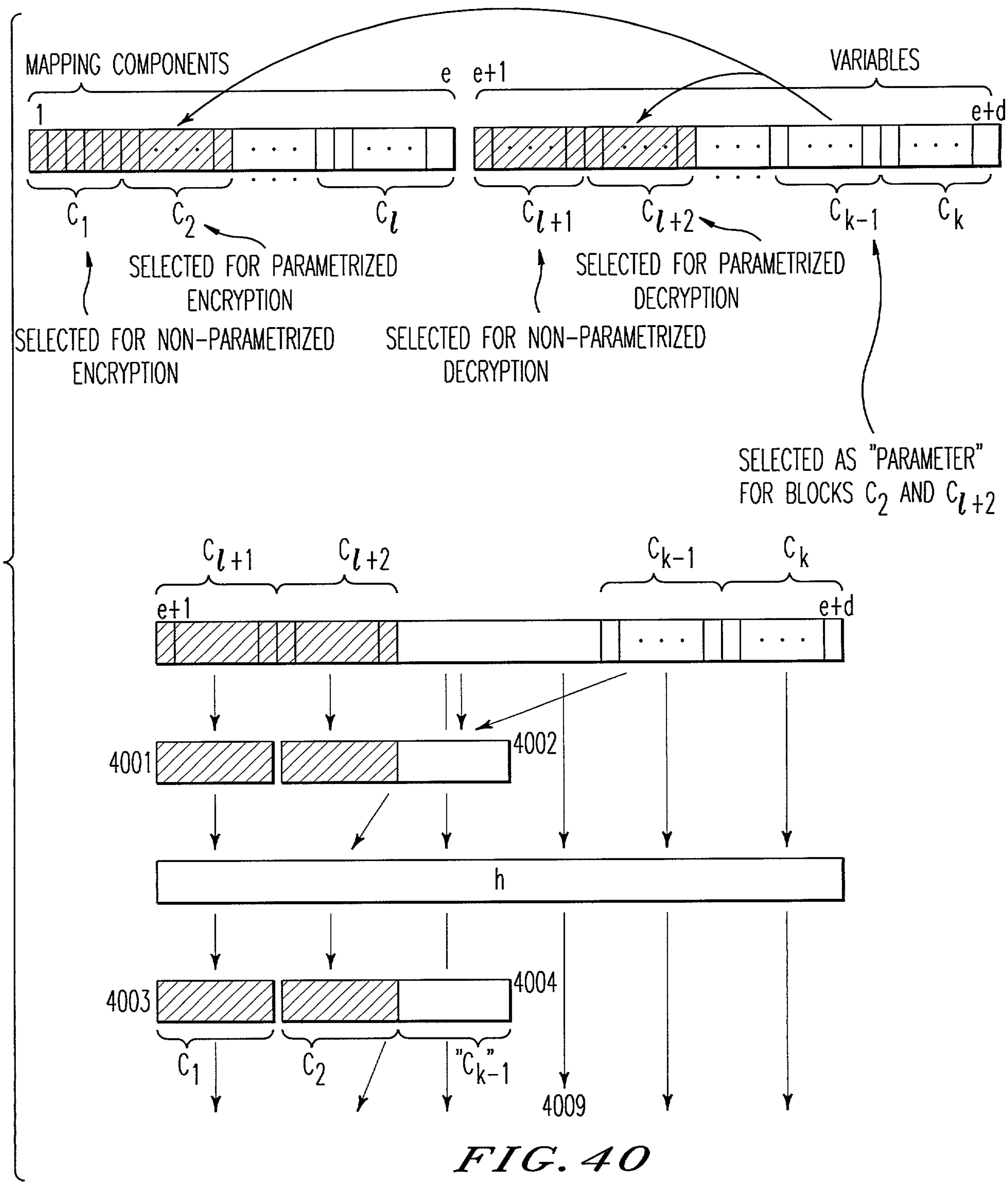
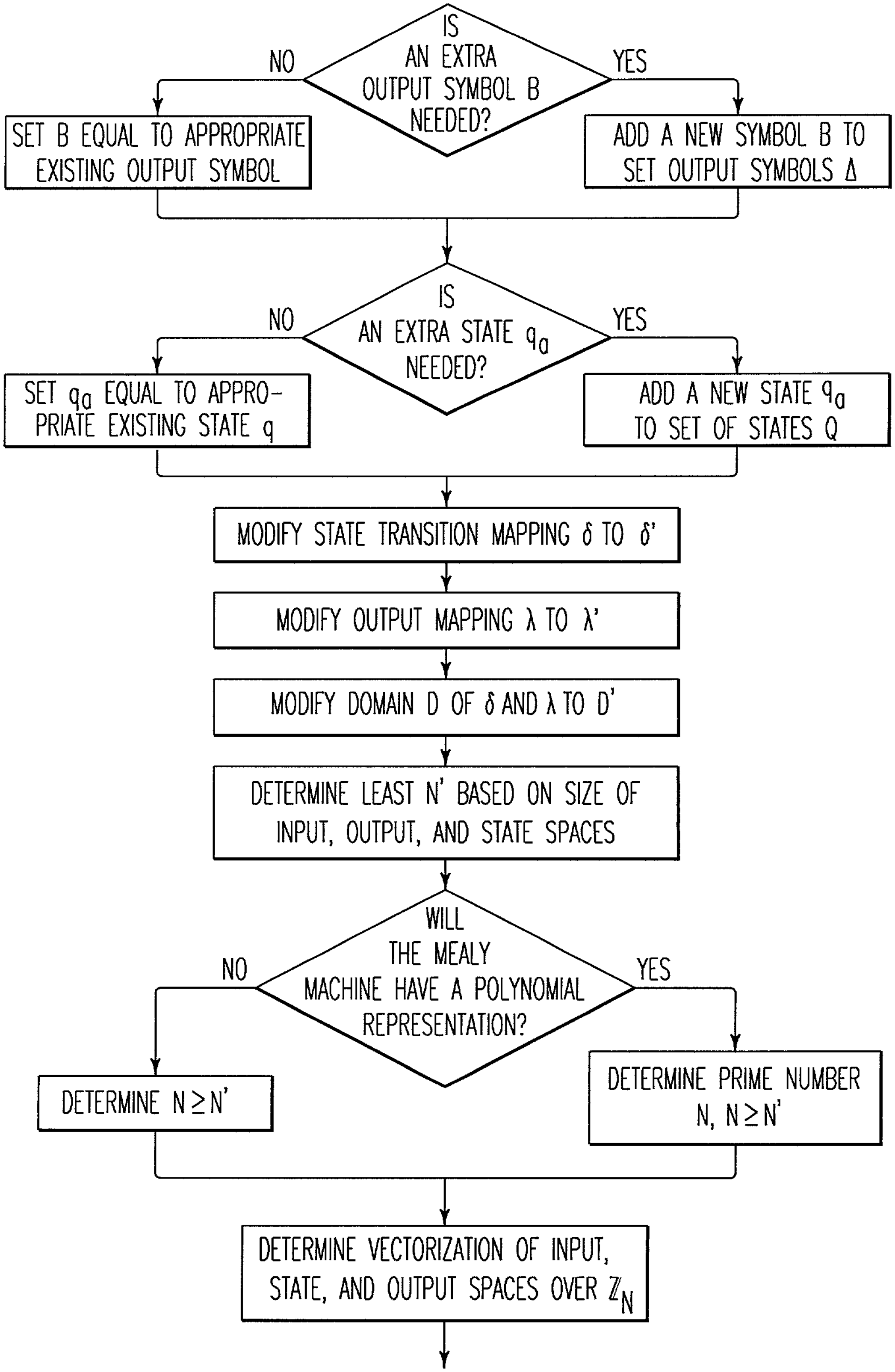


FIG. 39B



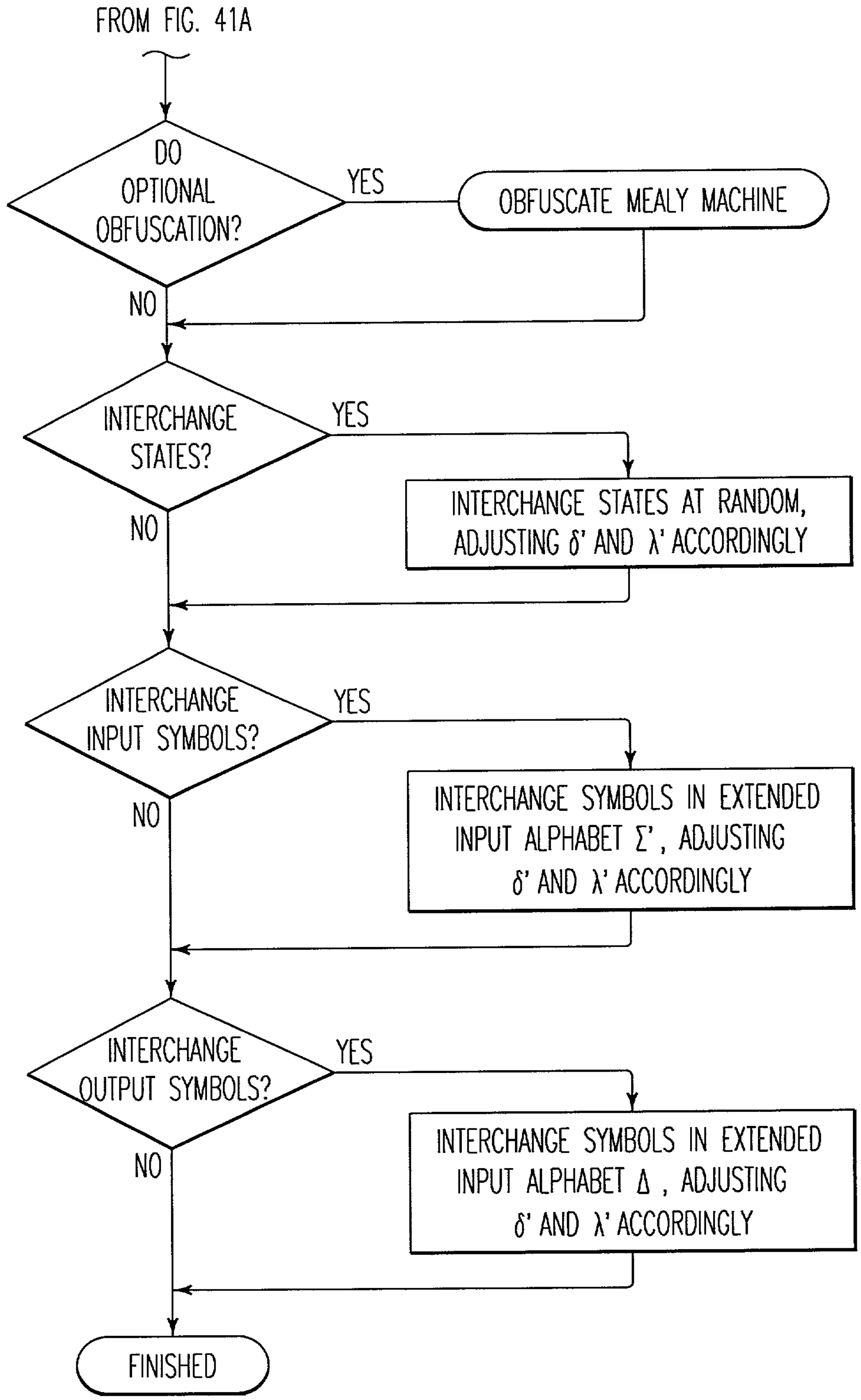
MAPPING COMPONENTS
1
e
e+1
VARIABLES
e+d
C_1
C_2
C_l
C_{l+1}
C_{l+2}
C_{k-1}
C_k
SELECTED FOR PARAMETRIZED ENCRYPTION
SELECTED FOR NON-PARAMETRIZED ENCRYPTION
SELECTED FOR PARAMETRIZED DECRYPTION
SELECTED FOR NON-PARAMETRIZED DECRYPTION
SELECTED AS "PARAMETER" FOR BLOCKS C_2 AND C_{l+2}
C_{l+1}
C_{l+2}
C_{k-1}
C_k
e+1
e+d
4001
4002
h
4003
4004
C_1
C_2
"C_{k-1}"
4009

FIG. 40



TO FIG. 41B

FIG. 41A



FROM FIG. 41A
DO OPTIONAL OBFUSCATION?
YES
OBFUSCATE MEALY MACHINE
NO
INTERCHANGE STATES?
YES
INTERCHANGE STATES AT RANDOM, ADJUSTING δ' AND λ' ACCORDINGLY
NO
INTERCHANGE INPUT SYMBOLS?
YES
INTERCHANGE SYMBOLS IN EXTENDED INPUT ALPHABET Σ', ADJUSTING δ' AND λ' ACCORDINGLY
NO
INTERCHANGE OUTPUT SYMBOLS?
YES
INTERCHANGE SYMBOLS IN EXTENDED INPUT ALPHABET Δ, ADJUSTING δ' AND λ' ACCORDINGLY
NO
FINISHED

FIG. 41B

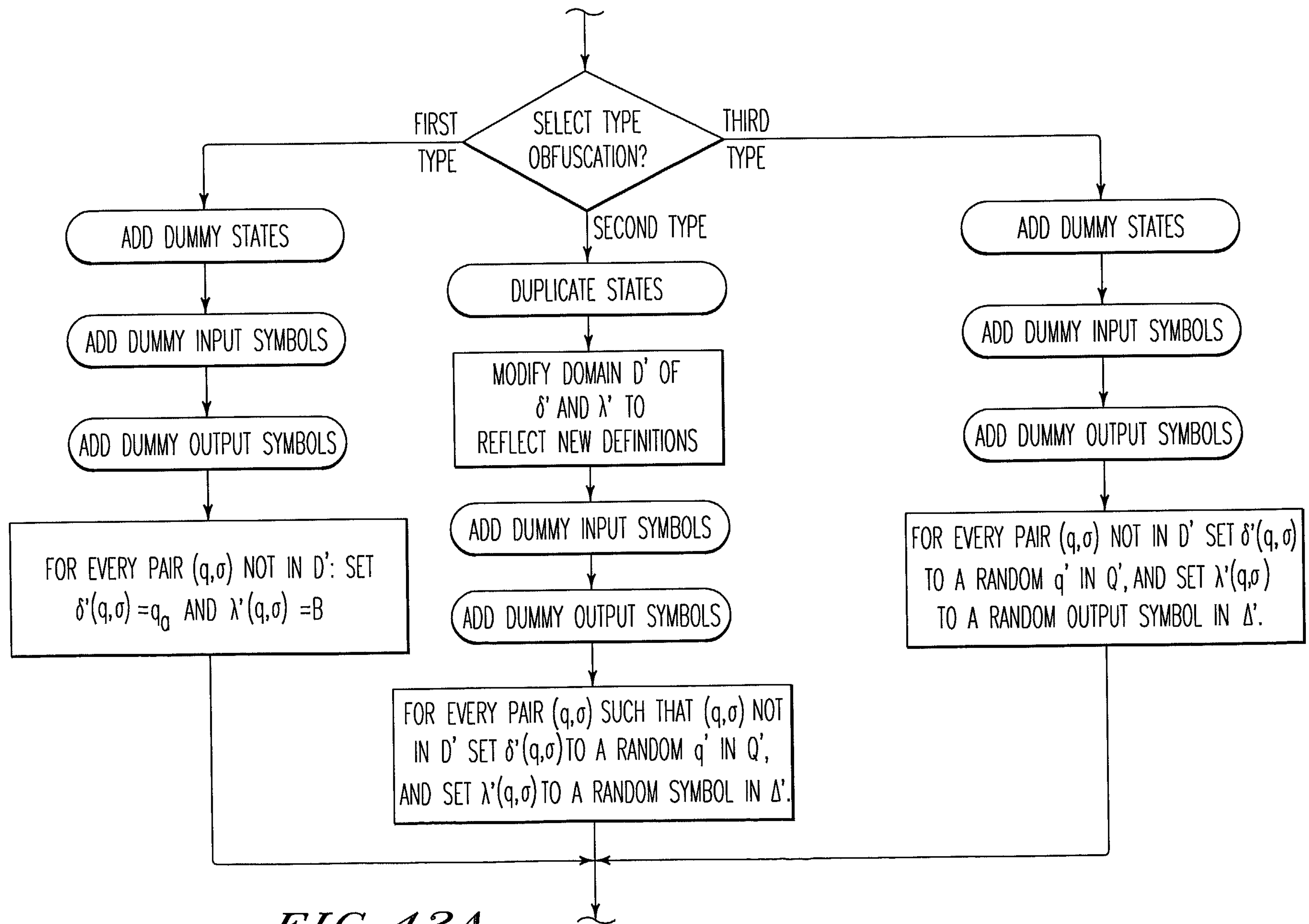

SELECT TYPE OBFUSCATION?
FIRST TYPE
SECOND TYPE
THIRD TYPE
ADD DUMMY STATES
ADD DUMMY INPUT SYMBOLS
ADD DUMMY OUTPUT SYMBOLS
FOR EVERY PAIR (q,σ) NOT IN D': SET δ'(q,σ) =q_a AND λ'(q,σ) =B
DUPLICATE STATES
MODIFY DOMAIN D' OF δ' AND λ' TO REFLECT NEW DEFINITIONS
ADD DUMMY INPUT SYMBOLS
ADD DUMMY OUTPUT SYMBOLS
FOR EVERY PAIR (q,σ) SUCH THAT (q,σ) NOT IN D' SET δ'(q,σ) TO A RANDOM q' IN Q', AND SET λ'(q,σ) TO A RANDOM SYMBOL IN Δ'.
ADD DUMMY STATES
ADD DUMMY INPUT SYMBOLS
ADD DUMMY OUTPUT SYMBOLS
FOR EVERY PAIR (q,σ) NOT IN D' SET δ'(q,σ) TO A RANDOM q' IN Q', AND SET λ'(q,σ) TO A RANDOM OUTPUT SYMBOL IN Δ'.

FIG. 42A

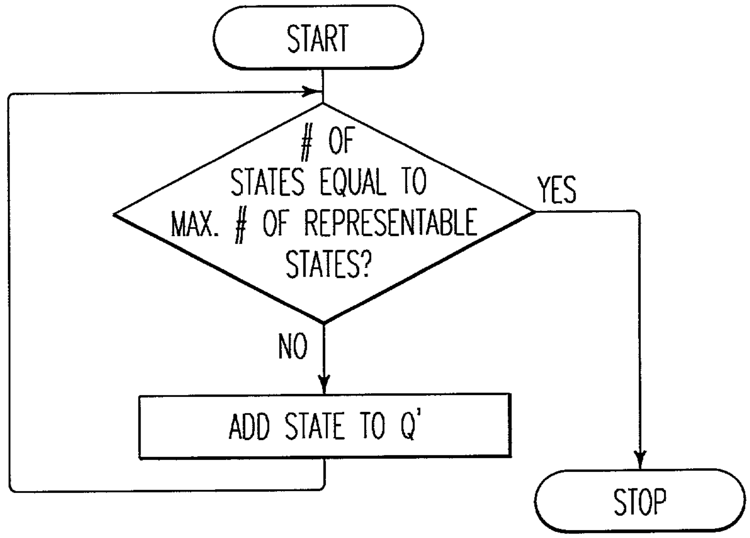


FIG. 42B

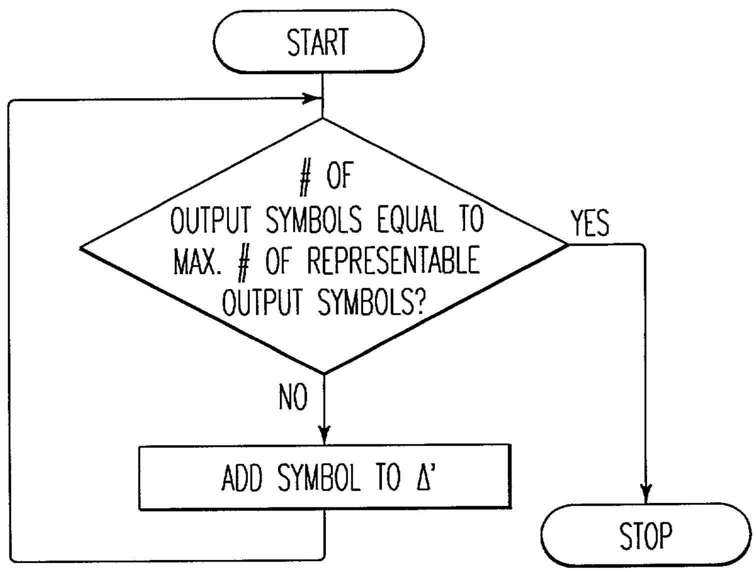


FIG. 42C

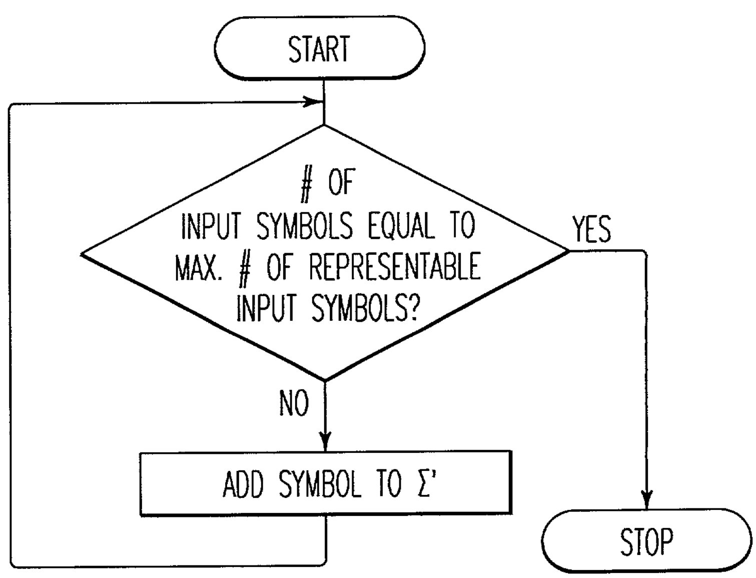
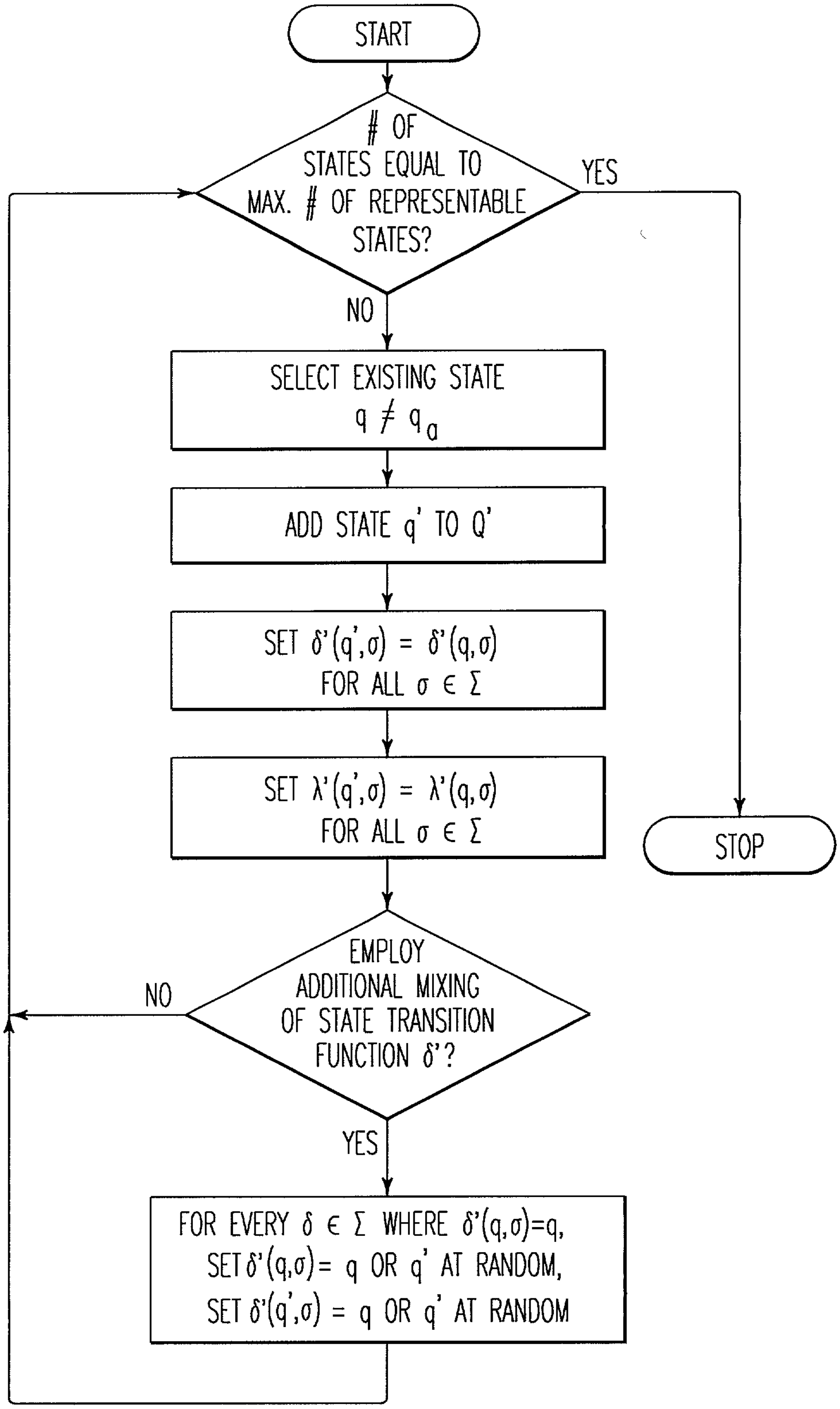


FIG. 42D



START
OF STATES EQUAL TO MAX. # OF REPRESENTABLE STATES?
YES
NO
SELECT EXISTING STATE $q \neq q_a$
ADD STATE q' TO Q'
SET $\delta'(q',\sigma) = \delta'(q,\sigma)$ FOR ALL $\sigma \in \Sigma$
SET $\lambda'(q',\sigma) = \lambda'(q,\sigma)$ FOR ALL $\sigma \in \Sigma$
STOP
EMPLOY ADDITIONAL MIXING OF STATE TRANSITION FUNCTION δ'?
NO
YES
FOR EVERY $\delta \in \Sigma$ WHERE $\delta'(q,\sigma)=q$, SET $\delta'(q,\sigma)$= q OR q' AT RANDOM, SET $\delta'(q',\sigma)$ = q OR q' AT RANDOM

FIG. 42E

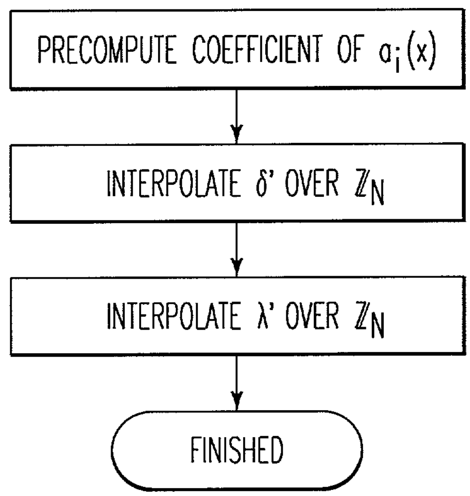


FIG. 43A

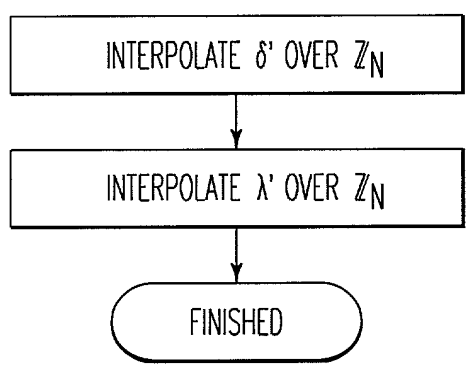


FIG. 43B

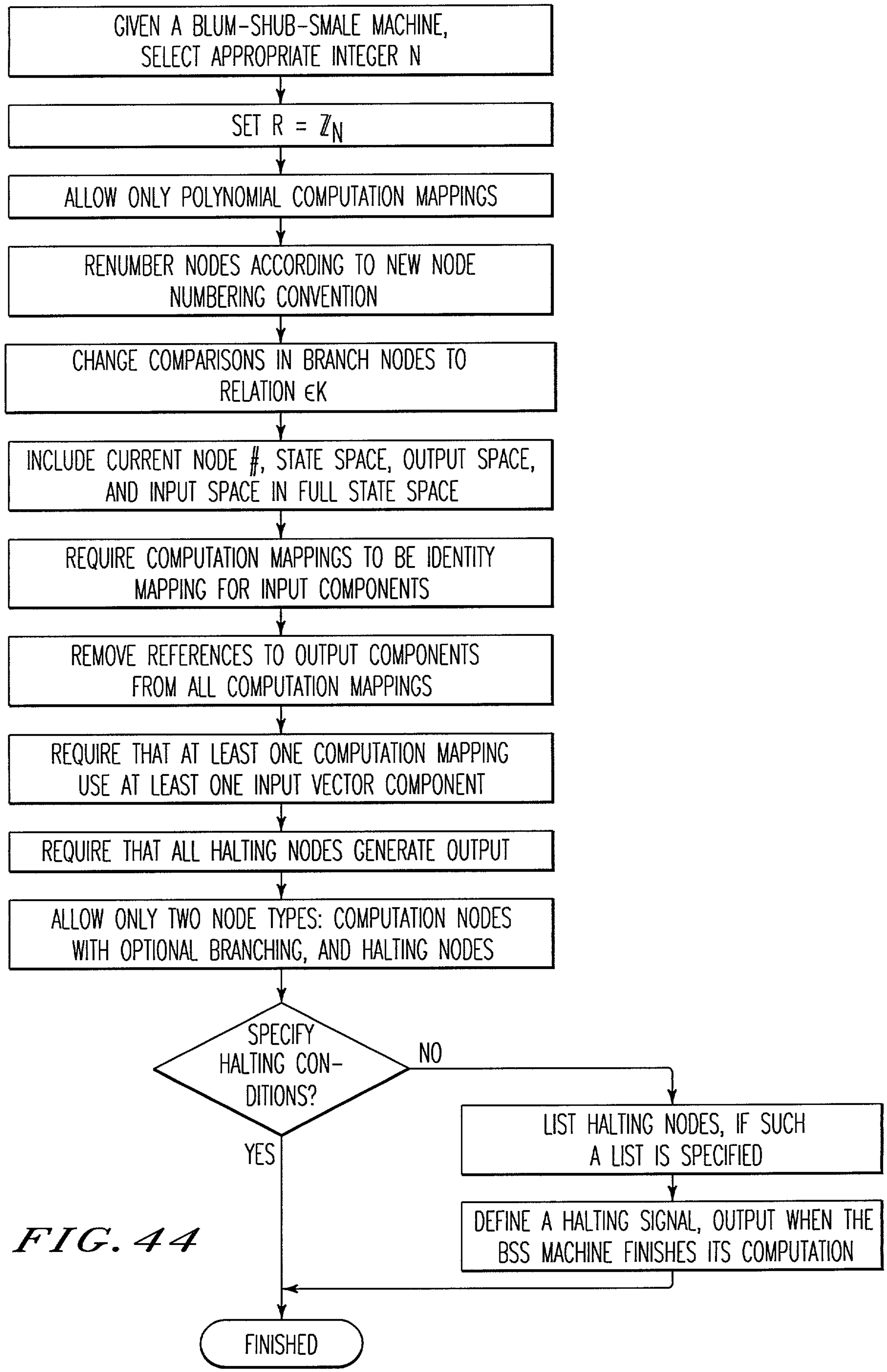

GIVEN A BLUM-SHUB-SMALE MACHINE, SELECT APPROPRIATE INTEGER N
SET R = $\mathbb{Z}_N$
ALLOW ONLY POLYNOMIAL COMPUTATION MAPPINGS
RENUMBER NODES ACCORDING TO NEW NODE NUMBERING CONVENTION
CHANGE COMPARISONS IN BRANCH NODES TO RELATION ϵK
INCLUDE CURRENT NODE #, STATE SPACE, OUTPUT SPACE, AND INPUT SPACE IN FULL STATE SPACE
REQUIRE COMPUTATION MAPPINGS TO BE IDENTITY MAPPING FOR INPUT COMPONENTS
REMOVE REFERENCES TO OUTPUT COMPONENTS FROM ALL COMPUTATION MAPPINGS
REQUIRE THAT AT LEAST ONE COMPUTATION MAPPING USE AT LEAST ONE INPUT VECTOR COMPONENT
REQUIRE THAT ALL HALTING NODES GENERATE OUTPUT
ALLOW ONLY TWO NODE TYPES: COMPUTATION NODES WITH OPTIONAL BRANCHING, AND HALTING NODES
SPECIFY HALTING CON-DITIONS?
NO
LIST HALTING NODES, IF SUCH A LIST IS SPECIFIED
DEFINE A HALTING SIGNAL, OUTPUT WHEN THE BSS MACHINE FINISHES ITS COMPUTATION
YES
FINISHED

FIG. 44

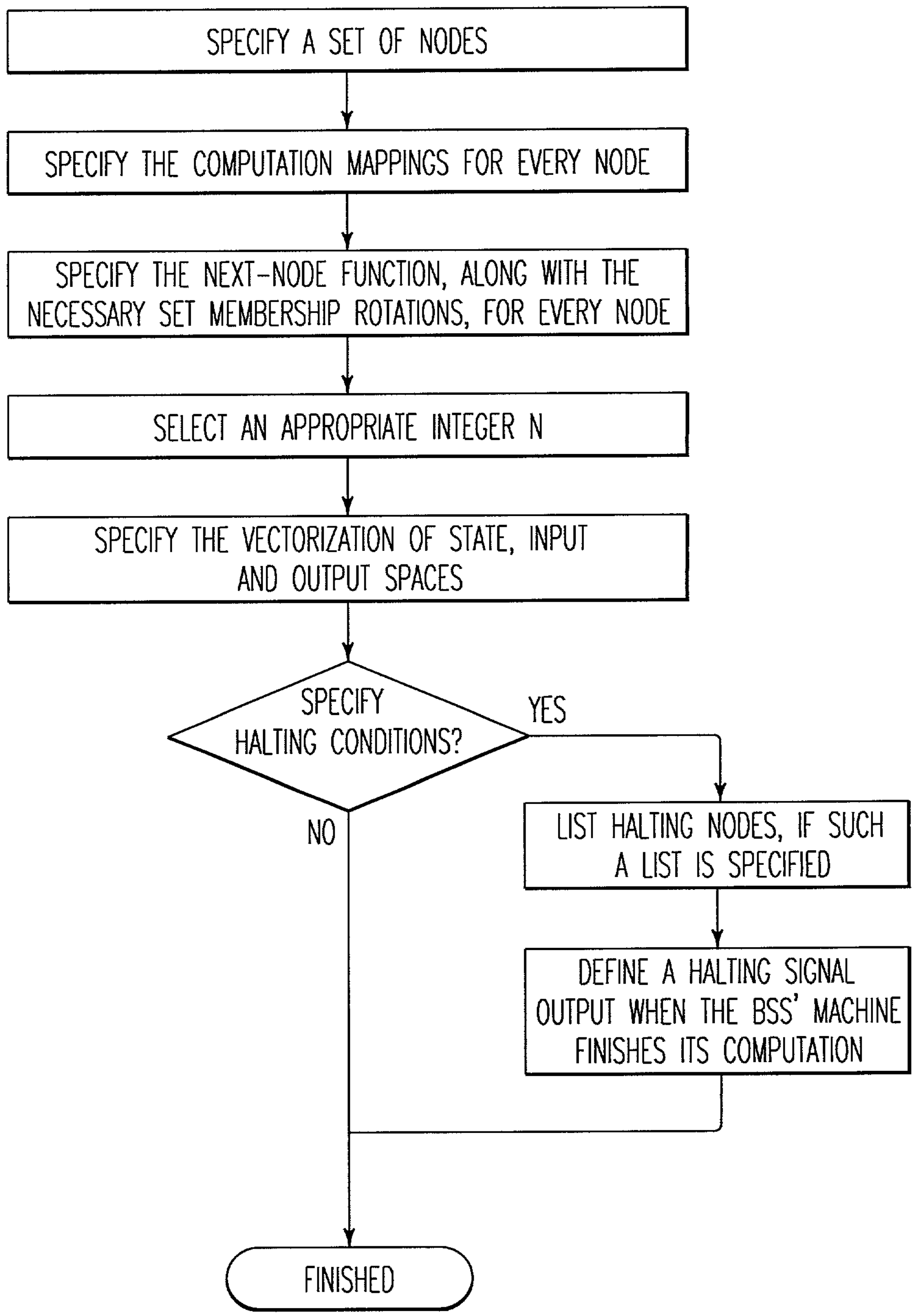

SPECIFY A SET OF NODES
SPECIFY THE COMPUTATION MAPPINGS FOR EVERY NODE
SPECIFY THE NEXT-NODE FUNCTION, ALONG WITH THE NECESSARY SET MEMBERSHIP ROTATIONS, FOR EVERY NODE
SELECT AN APPROPRIATE INTEGER N
SPECIFY THE VECTORIZATION OF STATE, INPUT AND OUTPUT SPACES
SPECIFY HALTING CONDITIONS?
YES
NO
LIST HALTING NODES, IF SUCH A LIST IS SPECIFIED
DEFINE A HALTING SIGNAL OUTPUT WHEN THE BSS' MACHINE FINISHES ITS COMPUTATION
FINISHED

FIG. 45

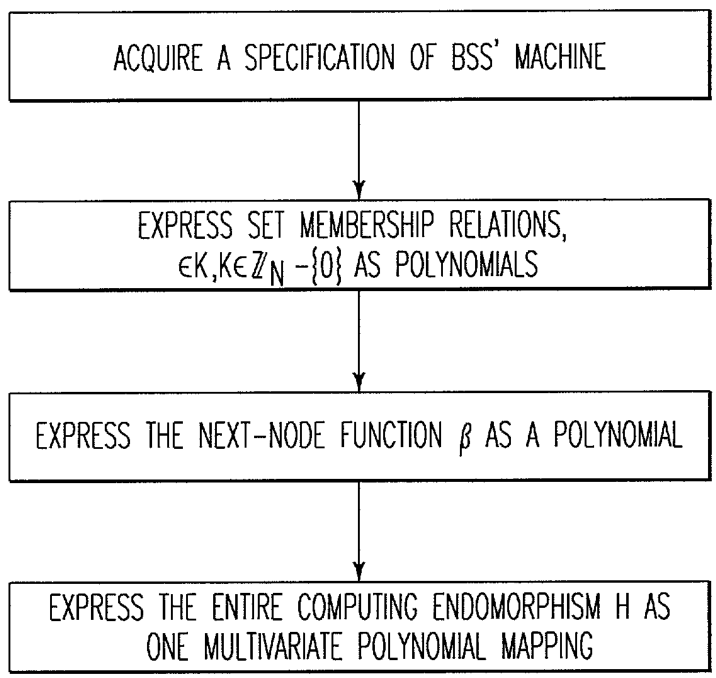

ACQUIRE A SPECIFICATION OF BSS' MACHINE
EXPRESS SET MEMBERSHIP RELATIONS, $\in K, K \in \mathbb{Z}_N - \{0\}$ AS POLYNOMIALS
EXPRESS THE NEXT-NODE FUNCTION β AS A POLYNOMIAL
EXPRESS THE ENTIRE COMPUTING ENDOMORPHISM H AS ONE MULTIVARIATE POLYNOMIAL MAPPING

FIG. 46

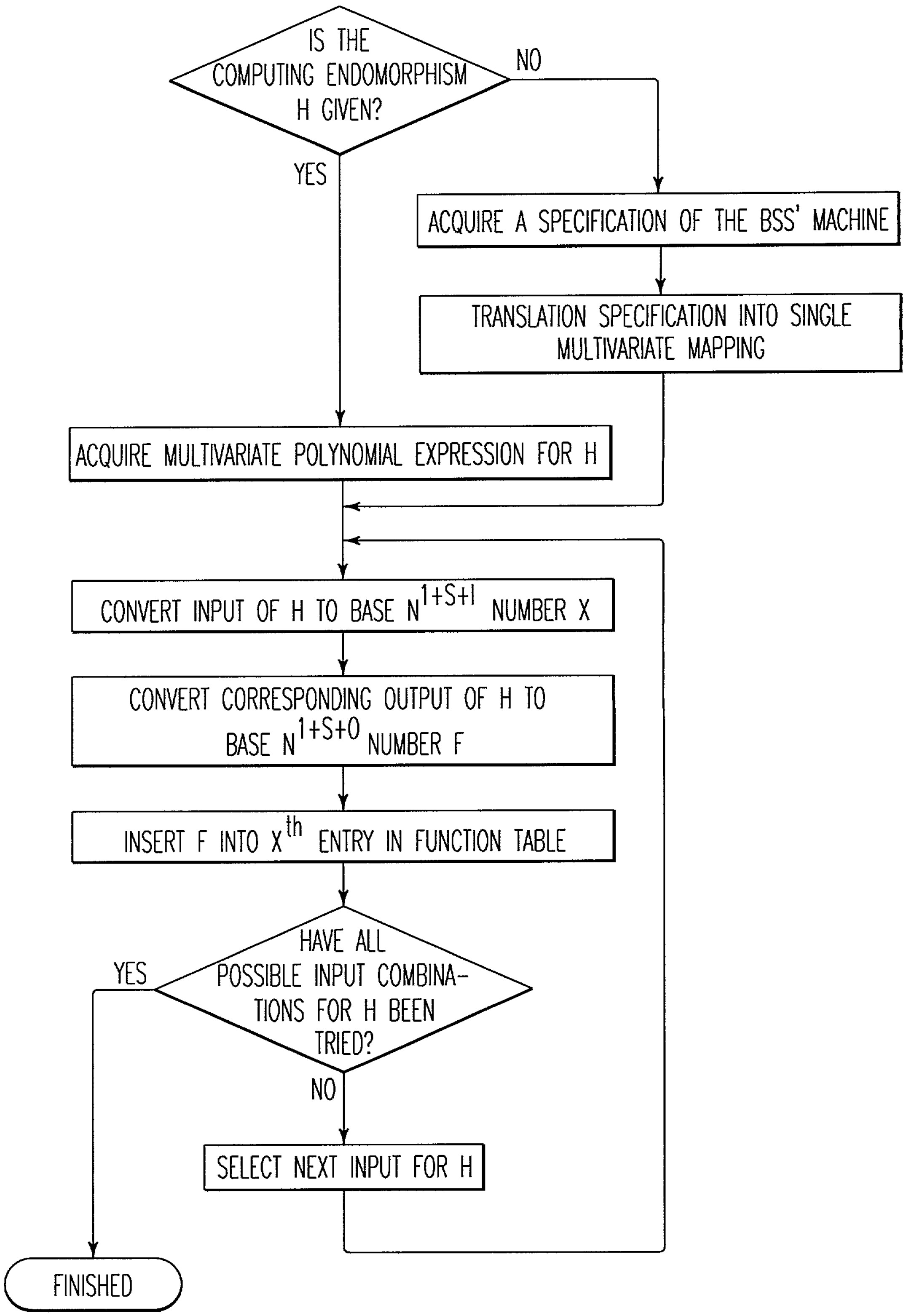

IS THE COMPUTING ENDOMORPHISM H GIVEN?
NO
YES
ACQUIRE A SPECIFICATION OF THE BSS' MACHINE
TRANSLATION SPECIFICATION INTO SINGLE MULTIVARIATE MAPPING
ACQUIRE MULTIVARIATE POLYNOMIAL EXPRESSION FOR H
CONVERT INPUT OF H TO BASE N^{1+S+I} NUMBER X
CONVERT CORRESPONDING OUTPUT OF H TO BASE N^{1+S+0} NUMBER F
INSERT F INTO X^{th} ENTRY IN FUNCTION TABLE
HAVE ALL POSSIBLE INPUT COMBINATIONS FOR H BEEN TRIED?
YES
NO
SELECT NEXT INPUT FOR H
FINISHED

FIG. 47

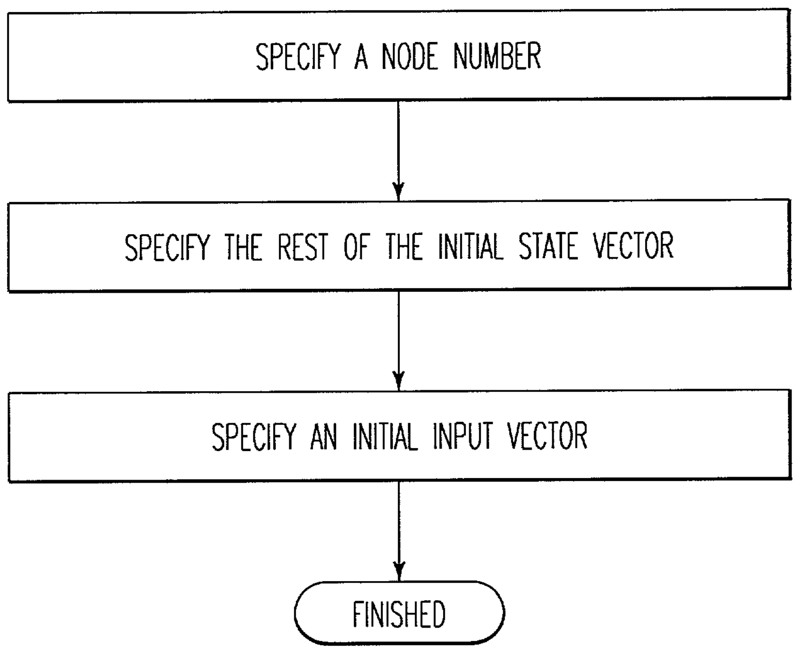

SPECIFY A NODE NUMBER
SPECIFY THE REST OF THE INITIAL STATE VECTOR
SPECIFY AN INITIAL INPUT VECTOR
FINISHED

FIG. 48

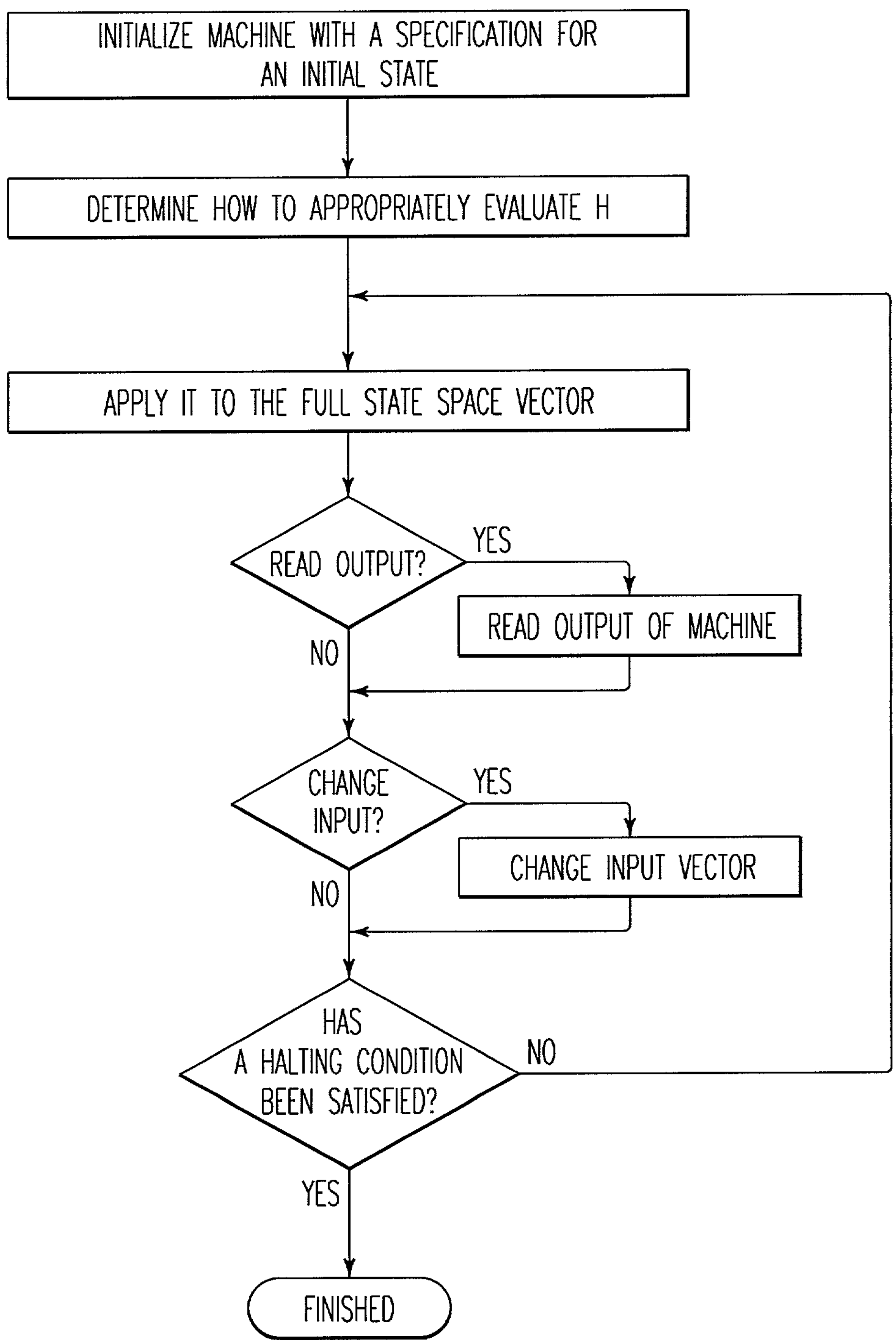

INITIALIZE MACHINE WITH A SPECIFICATION FOR AN INITIAL STATE
DETERMINE HOW TO APPROPRIATELY EVALUATE H
APPLY IT TO THE FULL STATE SPACE VECTOR
READ OUTPUT?
YES
READ OUTPUT OF MACHINE
NO
CHANGE INPUT?
YES
CHANGE INPUT VECTOR
NO
HAS A HALTING CONDITION BEEN SATISFIED?
NO
YES
FINISHED

FIG. 49

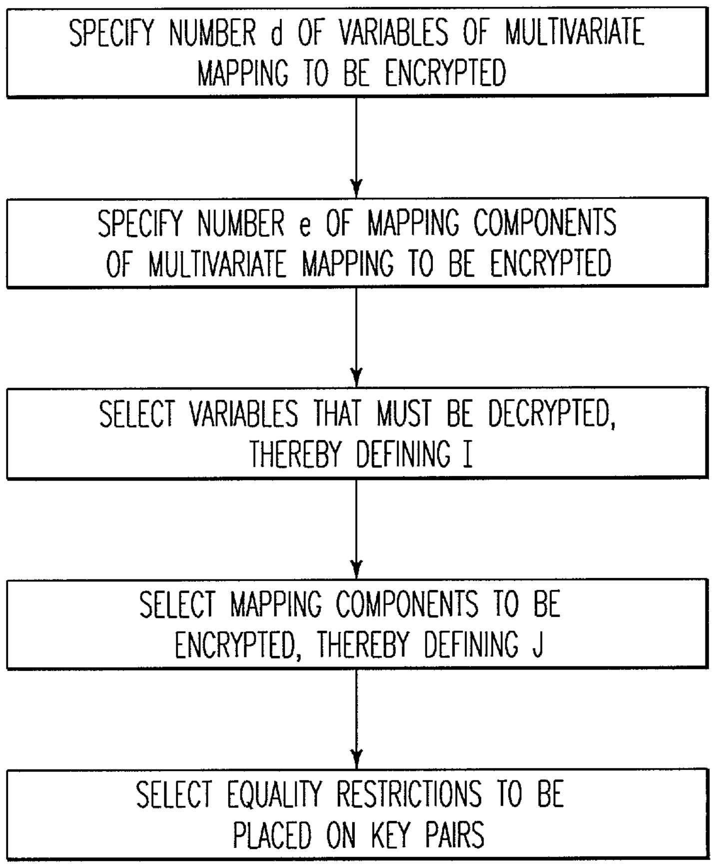
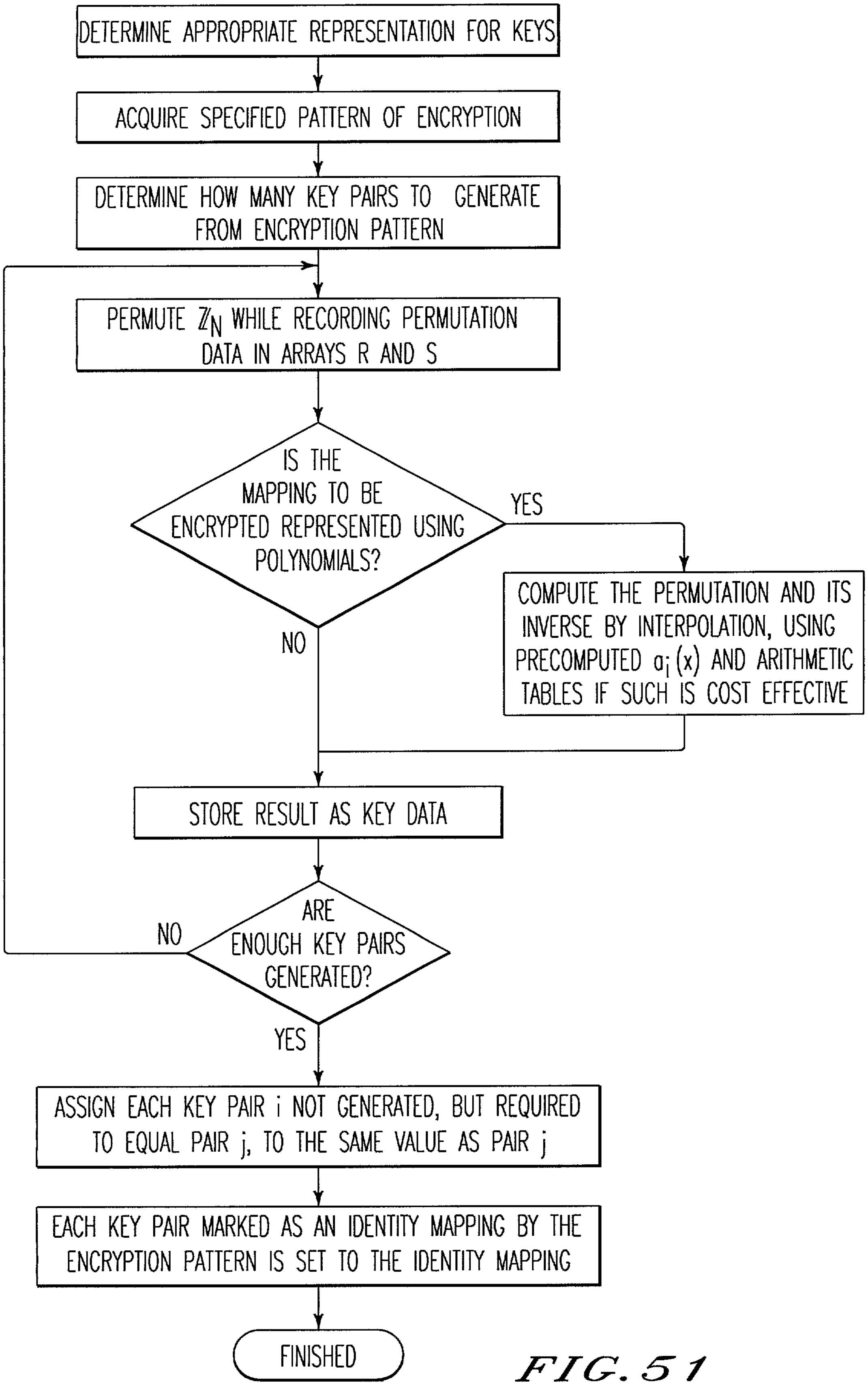

SPECIFY NUMBER d OF VARIABLES OF MULTIVARIATE MAPPING TO BE ENCRYPTED
SPECIFY NUMBER e OF MAPPING COMPONENTS OF MULTIVARIATE MAPPING TO BE ENCRYPTED
SELECT VARIABLES THAT MUST BE DECRYPTED, THEREBY DEFINING I
SELECT MAPPING COMPONENTS TO BE ENCRYPTED, THEREBY DEFINING J
SELECT EQUALITY RESTRICTIONS TO BE PLACED ON KEY PAIRS

FIG. 50



DETERMINE APPROPRIATE REPRESENTATION FOR KEYS
ACQUIRE SPECIFIED PATTERN OF ENCRYPTION
DETERMINE HOW MANY KEY PAIRS TO GENERATE FROM ENCRYPTION PATTERN
PERMUTE $\mathbb{Z}_N$ WHILE RECORDING PERMUTATION DATA IN ARRAYS R AND S
IS THE MAPPING TO BE ENCRYPTED REPRESENTED USING POLYNOMIALS?
YES
COMPUTE THE PERMUTATION AND ITS INVERSE BY INTERPOLATION, USING PRECOMPUTED $a_i(x)$ AND ARITHMETIC TABLES IF SUCH IS COST EFFECTIVE
NO
STORE RESULT AS KEY DATA
ARE ENOUGH KEY PAIRS GENERATED?
NO
YES
ASSIGN EACH KEY PAIR i NOT GENERATED, BUT REQUIRED TO EQUAL PAIR j, TO THE SAME VALUE AS PAIR j
EACH KEY PAIR MARKED AS AN IDENTITY MAPPING BY THE ENCRYPTION PATTERN IS SET TO THE IDENTITY MAPPING
FINISHED

FIG. 51

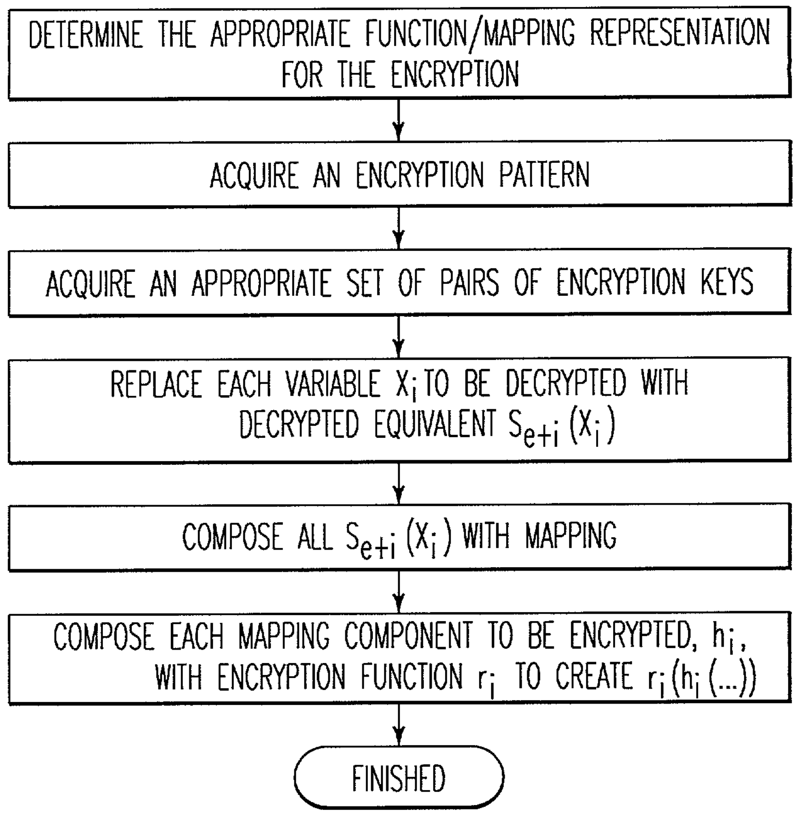

DETERMINE THE APPROPRIATE FUNCTION/MAPPING REPRESENTATION FOR THE ENCRYPTION
ACQUIRE AN ENCRYPTION PATTERN
ACQUIRE AN APPROPRIATE SET OF PAIRS OF ENCRYPTION KEYS
REPLACE EACH VARIABLE X_i TO BE DECRYPTED WITH DECRYPTED EQUIVALENT $S_{e+i}(X_i)$
COMPOSE ALL $S_{e+i}(X_i)$ WITH MAPPING
COMPOSE EACH MAPPING COMPONENT TO BE ENCRYPTED, h_i, WITH ENCRYPTION FUNCTION r_i TO CREATE $r_i(h_i(...))$
FINISHED

FIG. 52

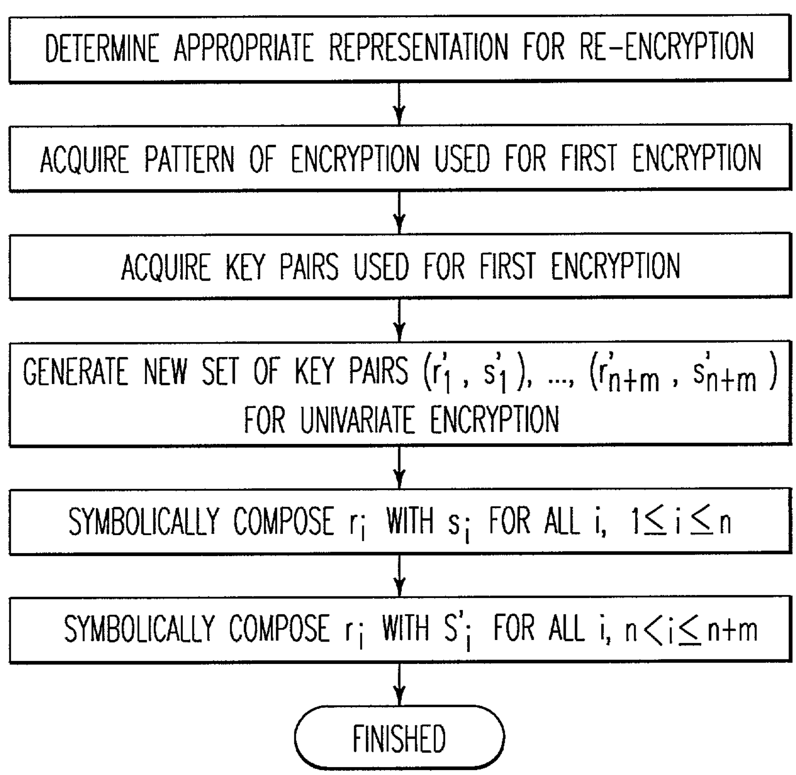

DETERMINE APPROPRIATE REPRESENTATION FOR RE-ENCRYPTION
ACQUIRE PATTERN OF ENCRYPTION USED FOR FIRST ENCRYPTION
ACQUIRE KEY PAIRS USED FOR FIRST ENCRYPTION
GENERATE NEW SET OF KEY PAIRS $(r'_1, s'_1), ..., (r'_{n+m}, s'_{n+m})$ FOR UNIVARIATE ENCRYPTION
SYMBOLICALLY COMPOSE r_i WITH s_i FOR ALL i, $1 \leq i \leq n$
SYMBOLICALLY COMPOSE r_i WITH S'_i FOR ALL i, $n < i \leq n+m$
FINISHED

FIG. 53

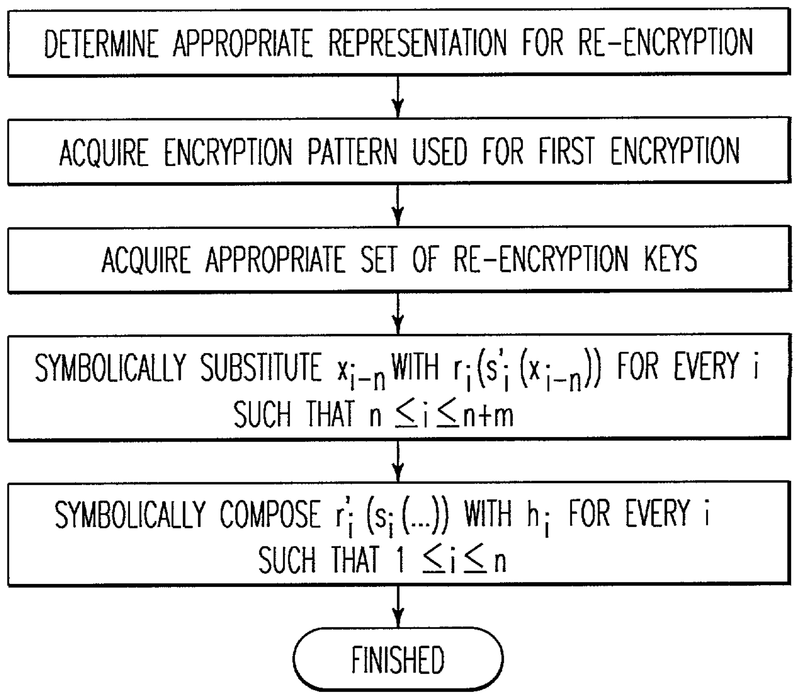

DETERMINE APPROPRIATE REPRESENTATION FOR RE-ENCRYPTION
ACQUIRE ENCRYPTION PATTERN USED FOR FIRST ENCRYPTION
ACQUIRE APPROPRIATE SET OF RE-ENCRYPTION KEYS
SYMBOLICALLY SUBSTITUTE x_{i-n} WITH $r_i(s'_i(x_{i-n}))$ FOR EVERY i SUCH THAT $n \leq i \leq n+m$
SYMBOLICALLY COMPOSE $r'_i(s_i(\ldots))$ WITH h_i FOR EVERY i SUCH THAT $1 \leq i \leq n$
FINISHED

FIG. 54

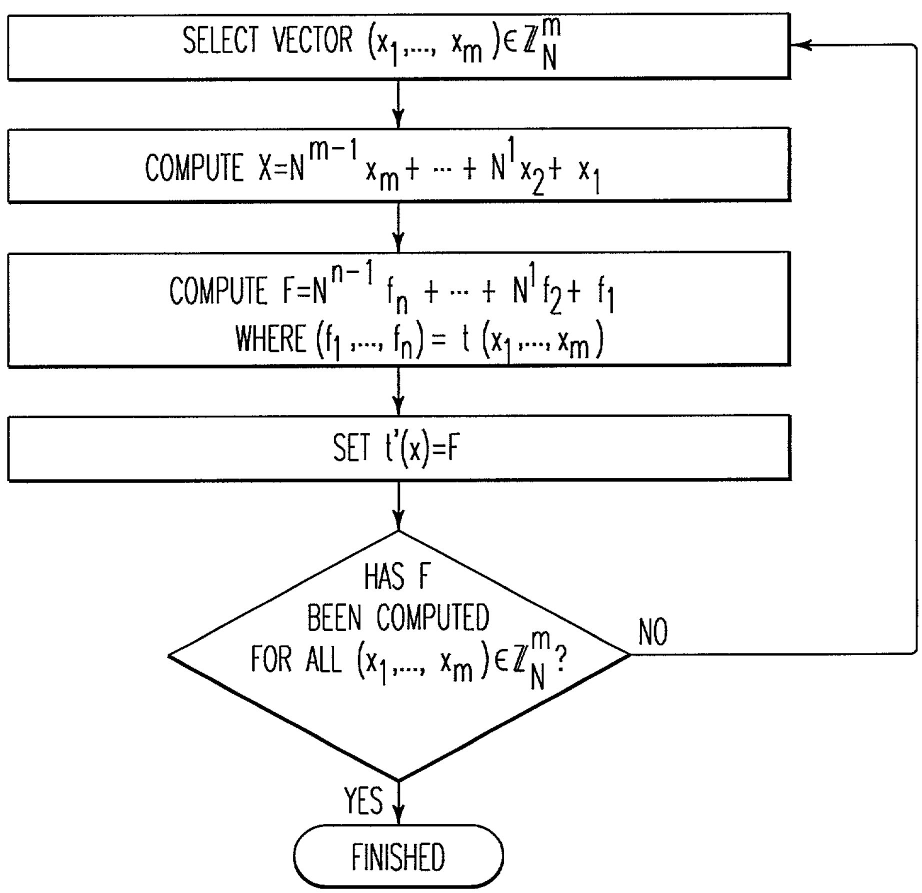

SELECT VECTOR $(x_1, \ldots, x_m) \in \mathbb{Z}_N^m$
COMPUTE $X = N^{m-1} x_m + \cdots + N^1 x_2 + x_1$
COMPUTE $F = N^{n-1} f_n + \cdots + N^1 f_2 + f_1$ WHERE $(f_1, \ldots, f_n) = t(x_1, \ldots, x_m)$
SET $t'(x) = F$
HAS F BEEN COMPUTED FOR ALL $(x_1, \ldots, x_m) \in \mathbb{Z}_N^m$?
NO
YES
FINISHED

FIG. 55

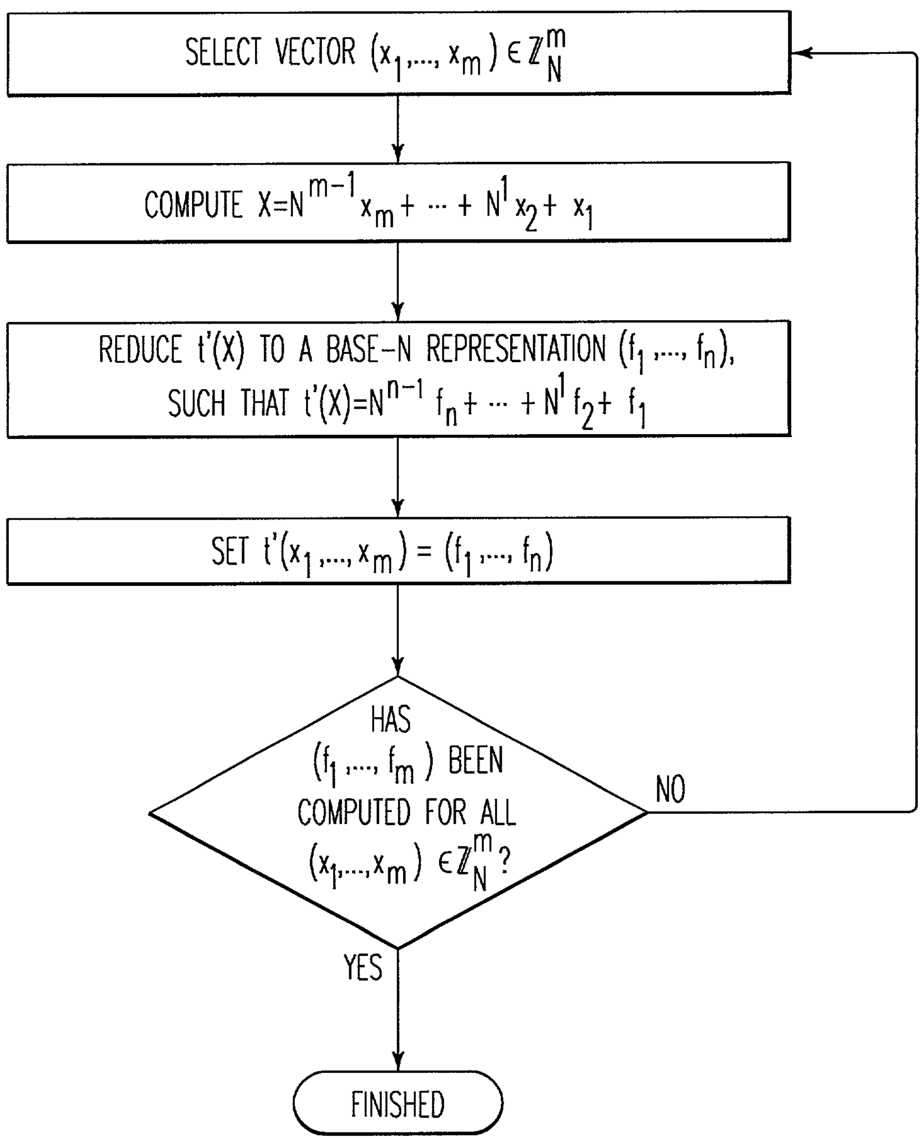

SELECT VECTOR $(x_1,\ldots,x_m)\in\mathbb{Z}_N^m$
COMPUTE $X=N^{m-1}x_m+\cdots+N^1x_2+x_1$
REDUCE $t'(X)$ TO A BASE-N REPRESENTATION $(f_1,\ldots,f_n)$, SUCH THAT $t'(X)=N^{n-1}f_n+\cdots+N^1f_2+f_1$
SET $t'(x_1,\ldots,x_m)=(f_1,\ldots,f_n)$
HAS $(f_1,\ldots,f_m)$ BEEN COMPUTED FOR ALL $(x_1,\ldots,x_m)\in\mathbb{Z}_N^m$?
NO
YES
FINISHED

FIG. 56

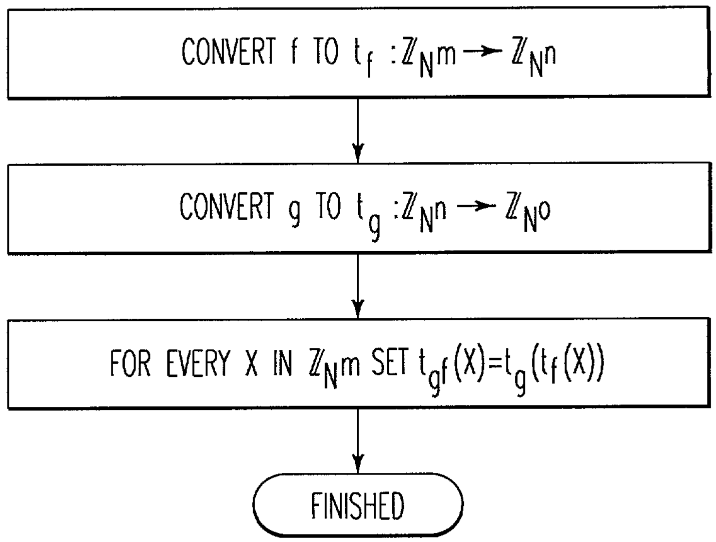

CONVERT f TO $t_f : \mathbb{Z}_N^m \rightarrow \mathbb{Z}_N^n$
CONVERT g TO $t_g : \mathbb{Z}_N^n \rightarrow \mathbb{Z}_N^o$
FOR EVERY X IN $\mathbb{Z}_N^m$ SET $t_{gf}(X)=t_g(t_f(X))$
FINISHED

FIG. 57

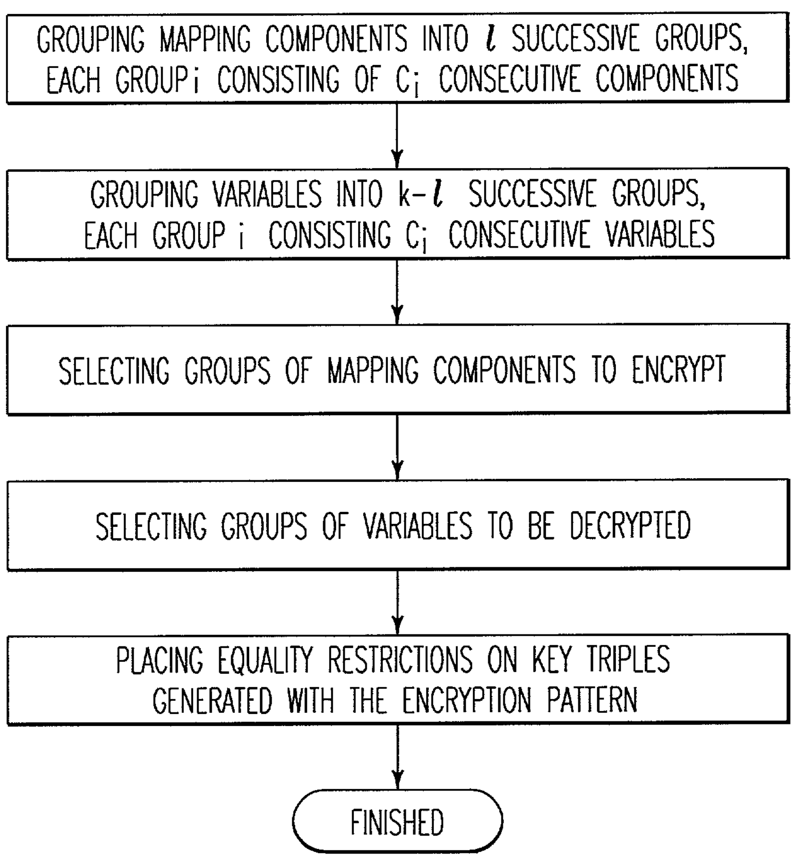

GROUPING MAPPING COMPONENTS INTO l SUCCESSIVE GROUPS, EACH GROUP i CONSISTING OF C_i CONSECUTIVE COMPONENTS
GROUPING VARIABLES INTO $k-l$ SUCCESSIVE GROUPS, EACH GROUP i CONSISTING C_i CONSECUTIVE VARIABLES
SELECTING GROUPS OF MAPPING COMPONENTS TO ENCRYPT
SELECTING GROUPS OF VARIABLES TO BE DECRYPTED
PLACING EQUALITY RESTRICTIONS ON KEY TRIPLES GENERATED WITH THE ENCRYPTION PATTERN
FINISHED

FIG. 58

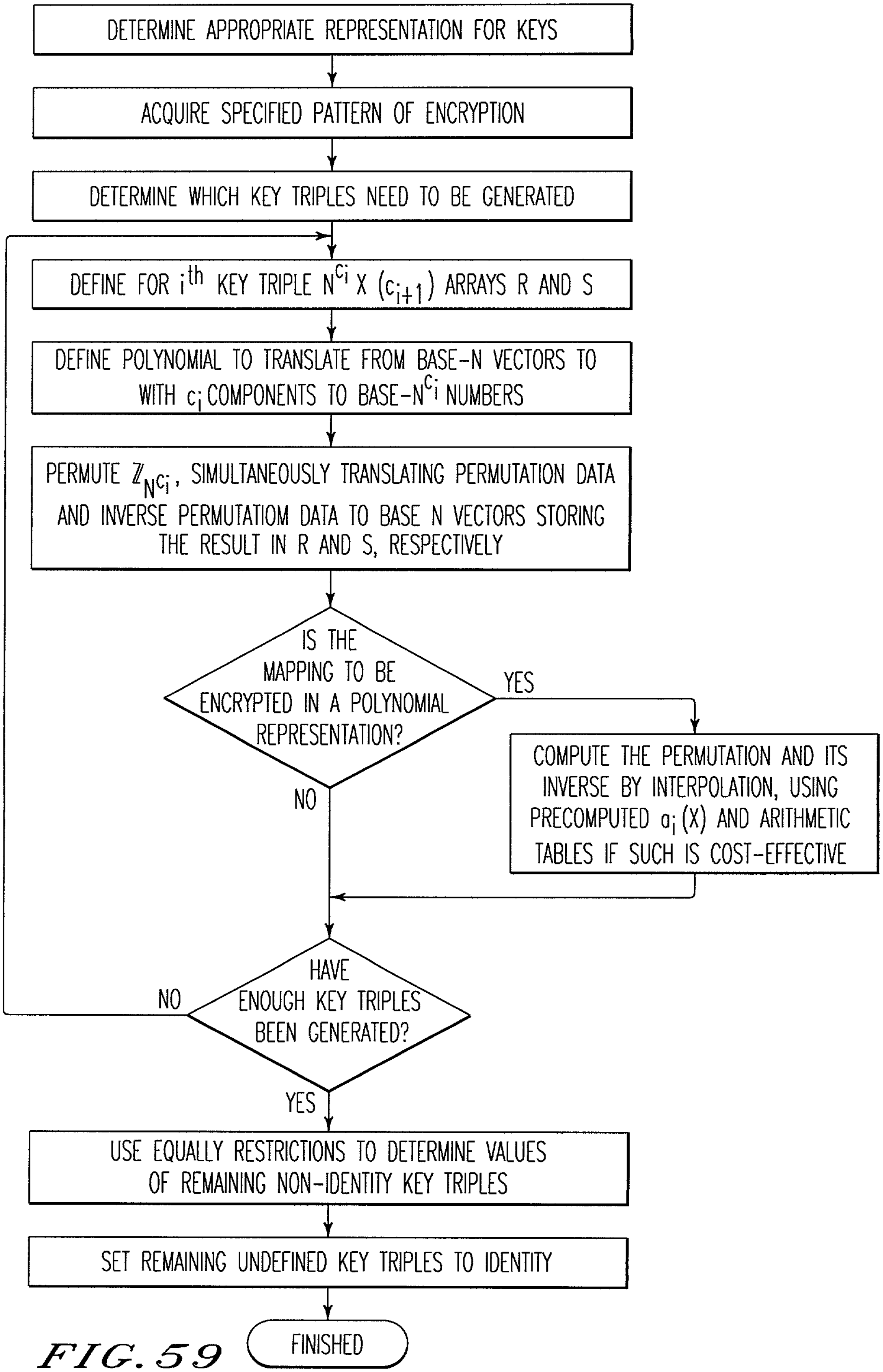

DETERMINE APPROPRIATE REPRESENTATION FOR KEYS
ACQUIRE SPECIFIED PATTERN OF ENCRYPTION
DETERMINE WHICH KEY TRIPLES NEED TO BE GENERATED
DEFINE FOR i^{th} KEY TRIPLE N^{c_i} X (c_{i+1}) ARRAYS R AND S
DEFINE POLYNOMIAL TO TRANSLATE FROM BASE-N VECTORS TO WITH c_i COMPONENTS TO BASE-N^{c_i} NUMBERS
PERMUTE $\mathbb{Z}_{N^{c_i}}$, SIMULTANEOUSLY TRANSLATING PERMUTATION DATA AND INVERSE PERMUTATIOM DATA TO BASE N VECTORS STORING THE RESULT IN R AND S, RESPECTIVELY
IS THE MAPPING TO BE ENCRYPTED IN A POLYNOMIAL REPRESENTATION?
YES
COMPUTE THE PERMUTATION AND ITS INVERSE BY INTERPOLATION, USING PRECOMPUTED $a_i(X)$ AND ARITHMETIC TABLES IF SUCH IS COST-EFFECTIVE
NO
HAVE ENOUGH KEY TRIPLES BEEN GENERATED?
NO
YES
USE EQUALLY RESTRICTIONS TO DETERMINE VALUES OF REMAINING NON-IDENTITY KEY TRIPLES
SET REMAINING UNDEFINED KEY TRIPLES TO IDENTITY
FINISHED

FIG. 59

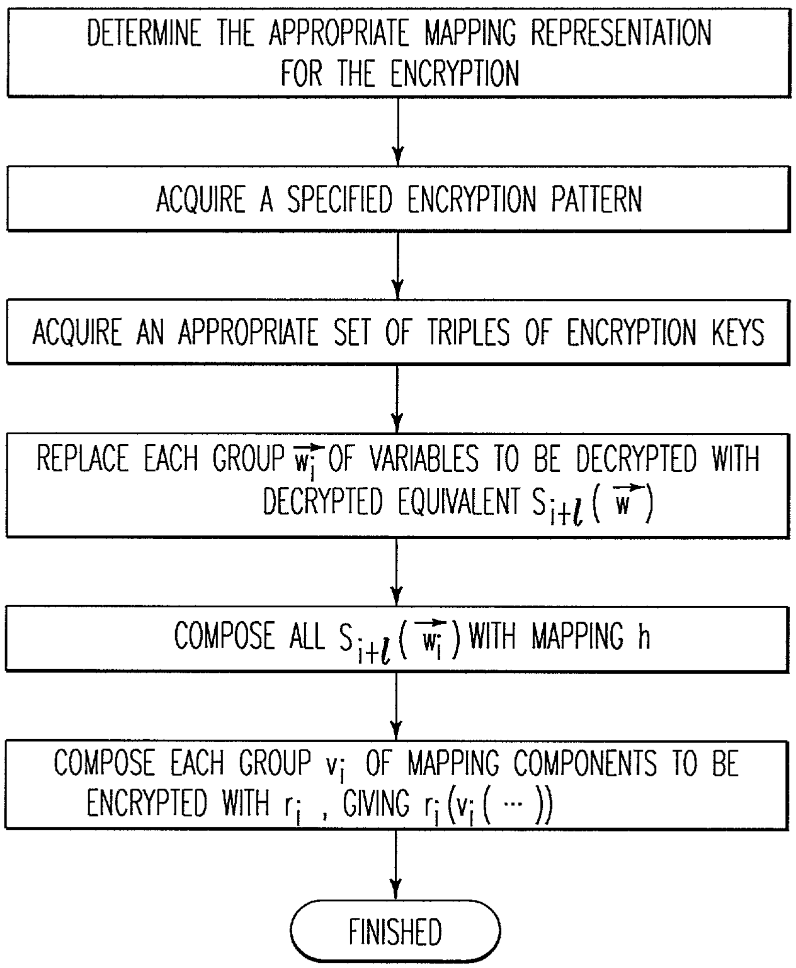

DETERMINE THE APPROPRIATE MAPPING REPRESENTATION FOR THE ENCRYPTION
ACQUIRE A SPECIFIED ENCRYPTION PATTERN
ACQUIRE AN APPROPRIATE SET OF TRIPLES OF ENCRYPTION KEYS
REPLACE EACH GROUP $\vec{w_i}$ OF VARIABLES TO BE DECRYPTED WITH DECRYPTED EQUIVALENT $S_{i+l}(\vec{w})$
COMPOSE ALL $S_{i+l}(\vec{w_i})$ WITH MAPPING h
COMPOSE EACH GROUP v_i OF MAPPING COMPONENTS TO BE ENCRYPTED WITH r_i , GIVING $r_i(v_i(\cdots))$
FINISHED

FIG. 60

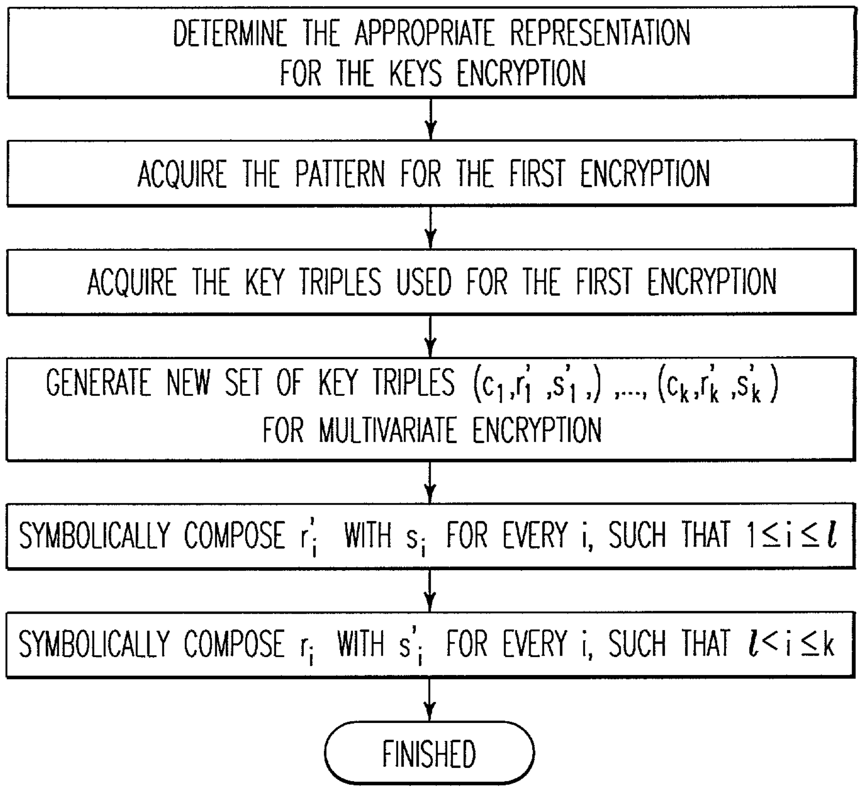

DETERMINE THE APPROPRIATE REPRESENTATION FOR THE KEYS ENCRYPTION
ACQUIRE THE PATTERN FOR THE FIRST ENCRYPTION
ACQUIRE THE KEY TRIPLES USED FOR THE FIRST ENCRYPTION
GENERATE NEW SET OF KEY TRIPLES $(c_1, r'_1, s'_1,), \ldots, (c_k, r'_k, s'_k)$ FOR MULTIVARIATE ENCRYPTION
SYMBOLICALLY COMPOSE r'_i WITH s_i FOR EVERY i, SUCH THAT $1 \le i \le l$
SYMBOLICALLY COMPOSE r_i WITH s'_i FOR EVERY i, SUCH THAT $l < i \le k$
FINISHED

FIG. 61

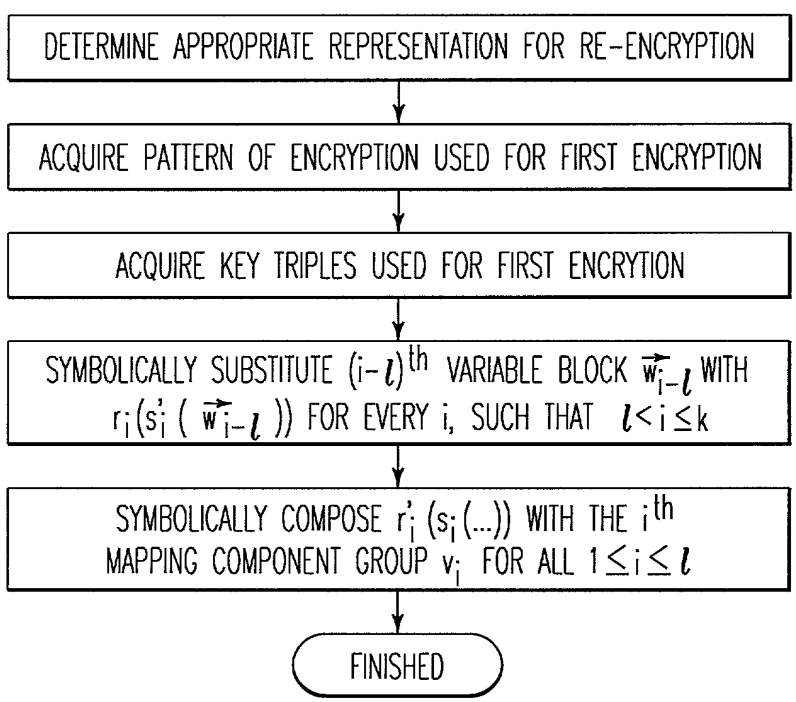

DETERMINE APPROPRIATE REPRESENTATION FOR RE-ENCRYPTION
ACQUIRE PATTERN OF ENCRYPTION USED FOR FIRST ENCRYPTION
ACQUIRE KEY TRIPLES USED FOR FIRST ENCRYTION
SYMBOLICALLY SUBSTITUTE $(i-l)^{th}$ VARIABLE BLOCK $\vec{w}_{i-l}$ WITH $r_i(s'_i(\vec{w}_{i-l}))$ FOR EVERY i, SUCH THAT $l < i \le k$
SYMBOLICALLY COMPOSE $r'_i(s_i(\ldots))$ WITH THE i^{th} MAPPING COMPONENT GROUP v_i FOR ALL $1 \le i \le l$
FINISHED

FIG. 62

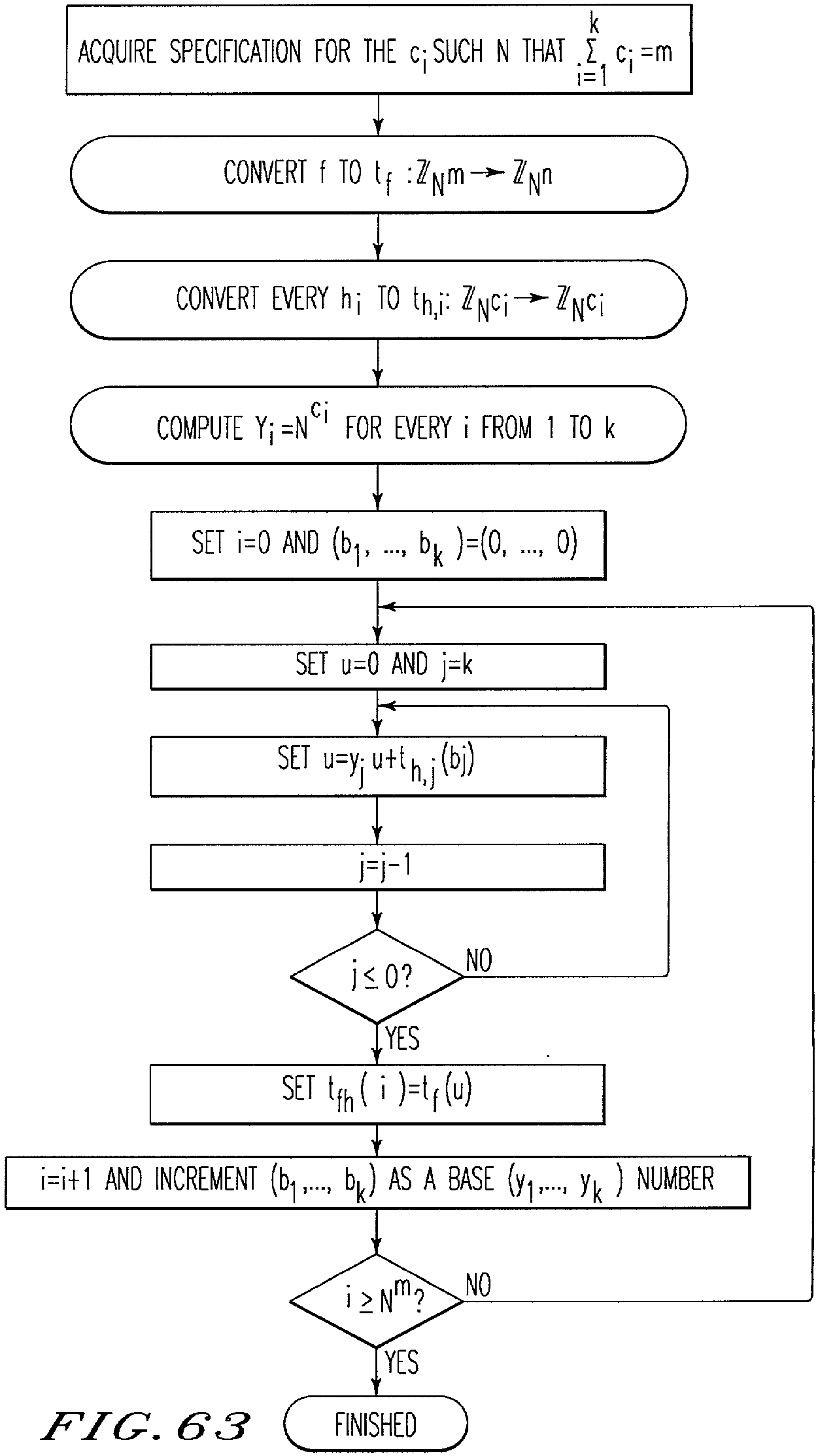

ACQUIRE SPECIFICATION FOR THE c_i SUCH N THAT $\sum_{i=1}^{k} c_i = m$
CONVERT f TO $t_f : \mathbb{Z}_N m \rightarrow \mathbb{Z}_N n$
CONVERT EVERY h_i TO $t_{h,i}: \mathbb{Z}_N c_i \rightarrow \mathbb{Z}_N c_i$
COMPUTE $Y_i = N^{c_i}$ FOR EVERY i FROM 1 TO k
SET i=0 AND $(b_1, ..., b_k)=(0, ..., 0)$
SET u=0 AND j=k
SET $u=y_j u+t_{h,j}(bj)$
j=j-1
$j \leq 0$?
NO
YES
SET $t_{fh}(i)=t_f(u)$
i=i+1 AND INCREMENT $(b_1,..., b_k)$ AS A BASE $(y_1,..., y_k)$ NUMBER
$i \geq N^m$?
NO
YES
FINISHED

FIG. 63

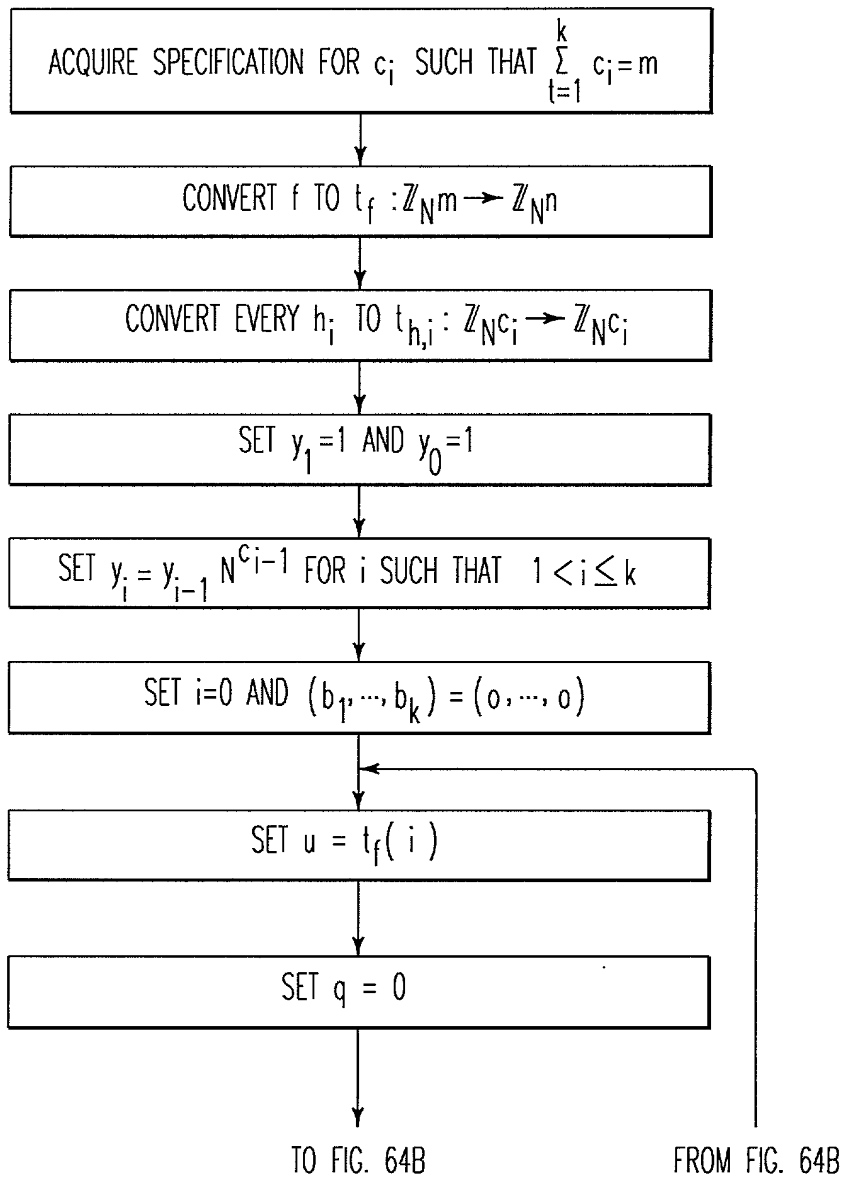

ACQUIRE SPECIFICATION FOR c_i SUCH THAT $\sum_{t=1}^{k} c_i = m$
CONVERT f TO $t_f : \mathbb{Z}_N m \rightarrow \mathbb{Z}_N n$
CONVERT EVERY h_i TO $t_{h,i} : \mathbb{Z}_N c_i \rightarrow \mathbb{Z}_N c_i$
SET $y_1 = 1$ AND $y_0 = 1$
SET $y_i = y_{i-1} N^{c_{i-1}}$ FOR i SUCH THAT $1 < i \leq k$
SET i=0 AND $(b_1, \cdots, b_k) = (o, \cdots, o)$
SET $u = t_f(i)$
SET q = 0
TO FIG. 64B
FROM FIG. 64B

FIG. 64A

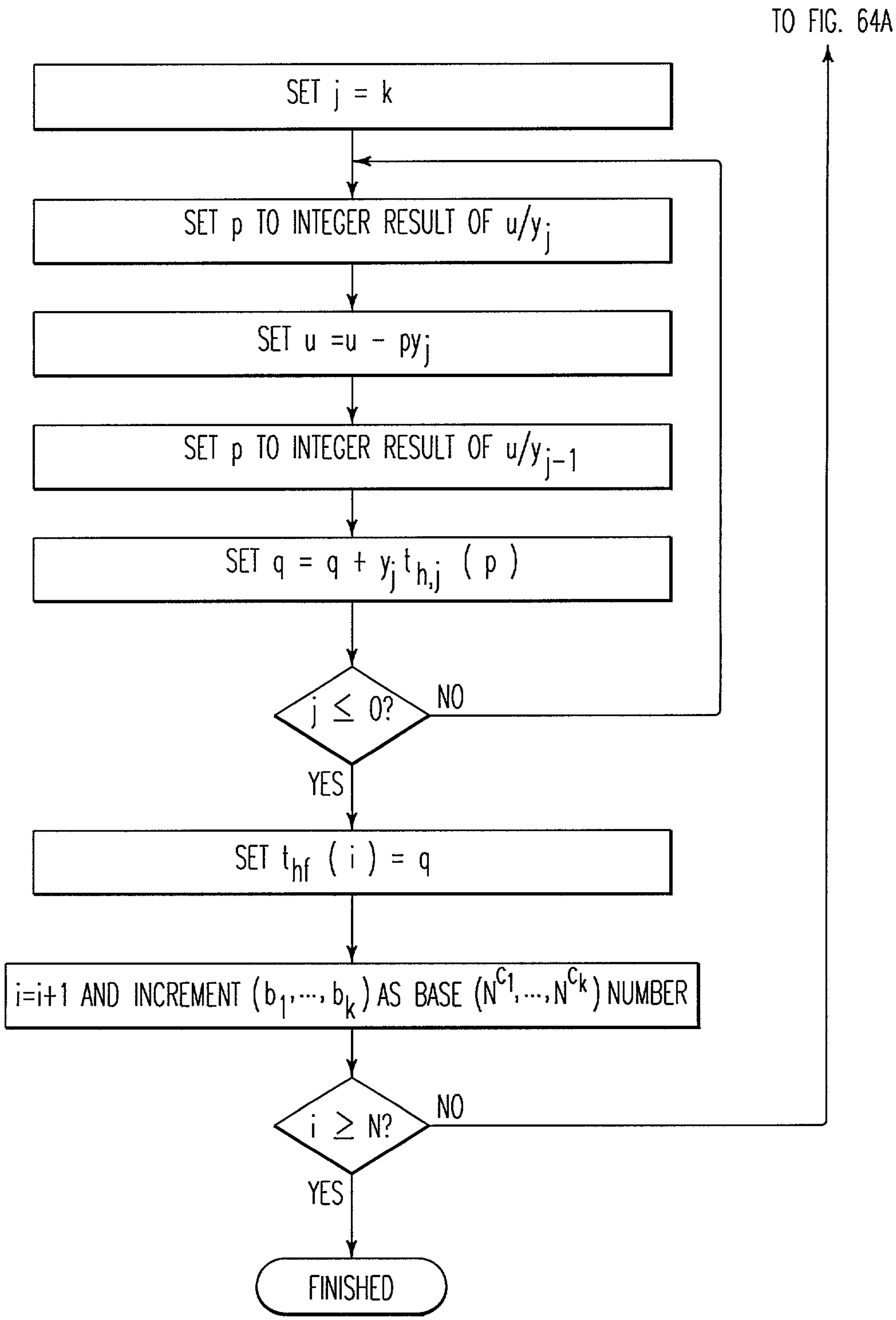

TO FIG. 64A
SET j = k
SET p TO INTEGER RESULT OF u/y_j
SET u =u − py_j
SET p TO INTEGER RESULT OF u/y_{j-1}
SET q = q + $y_j t_{h,j}$ (p)
$j \leq 0$?
NO
YES
SET t_{hf} (i) = q
i=i+1 AND INCREMENT $(b_1,\ldots,b_k)$ AS BASE $(N^{c_1},\ldots,N^{c_k})$ NUMBER
$i \geq N$?
NO
YES
FINISHED

FIG. 64B

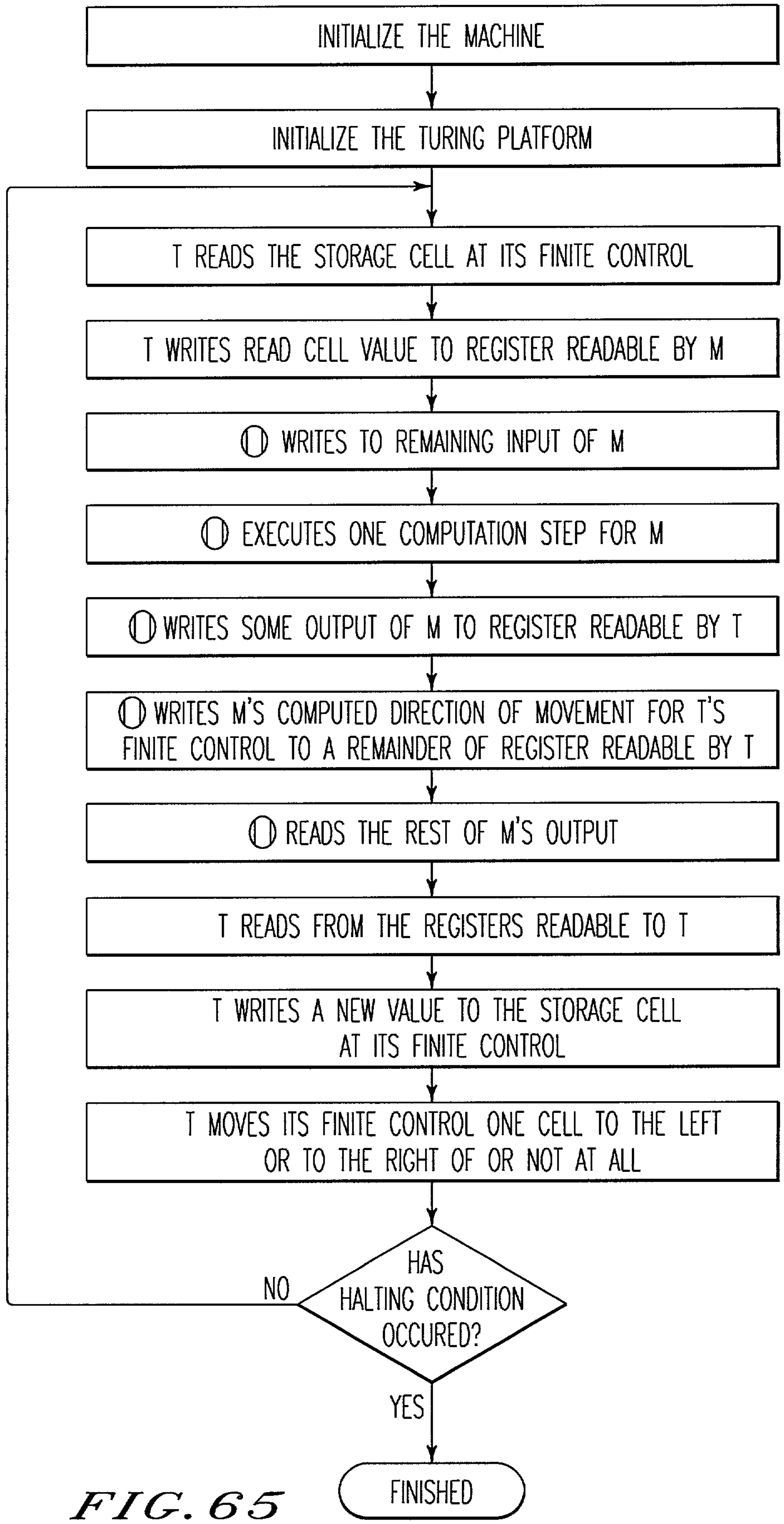
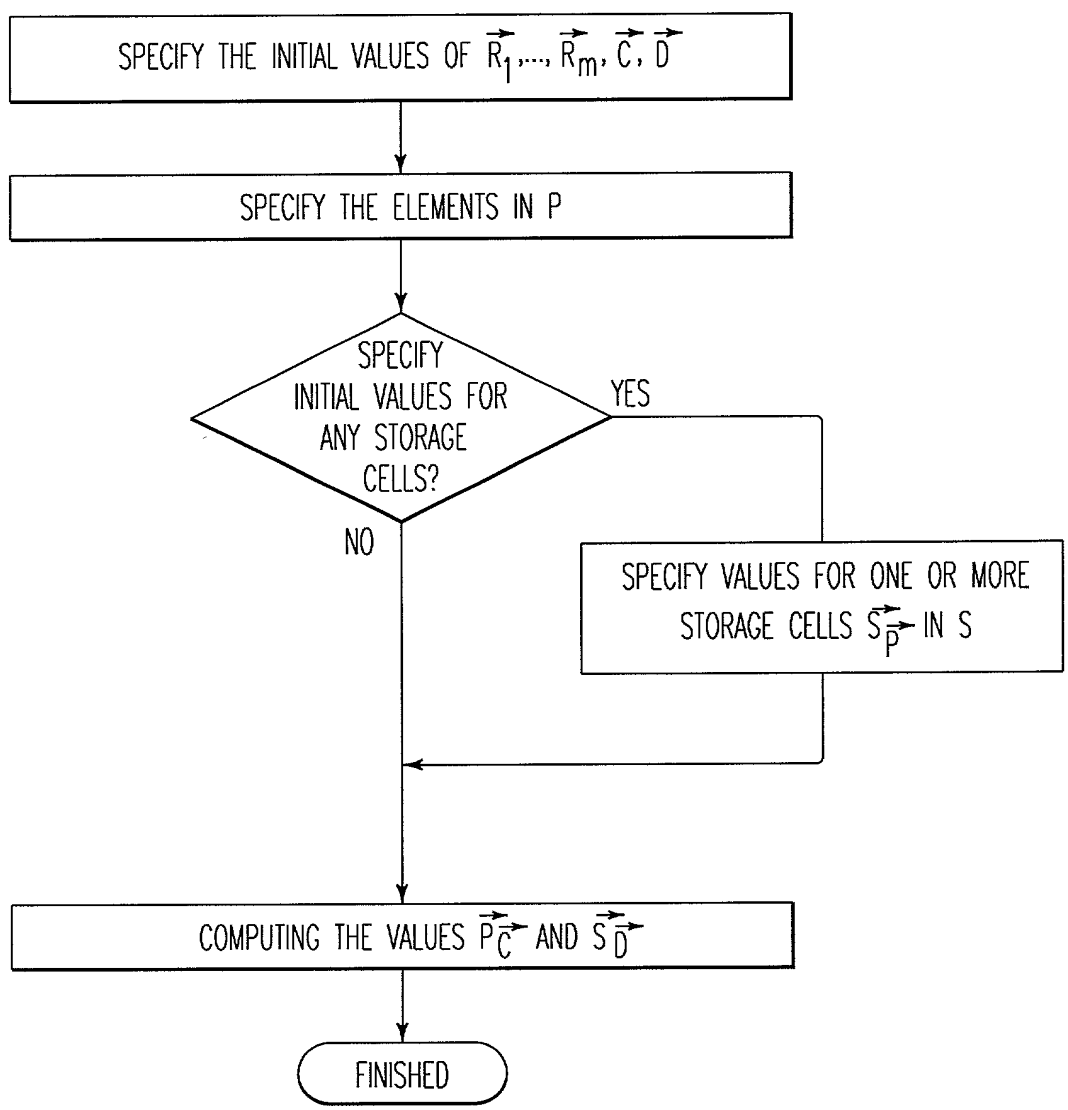
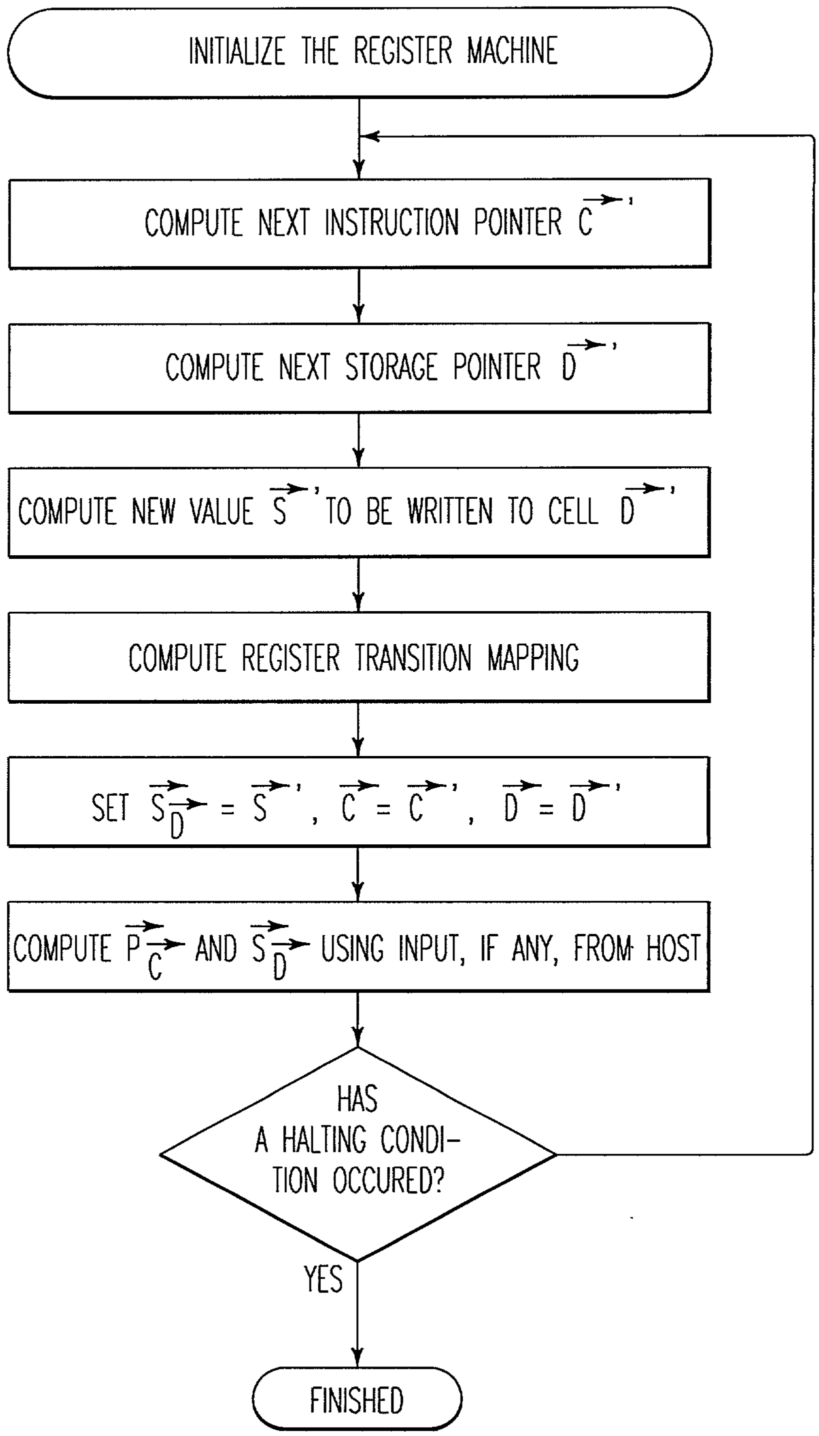

INITIALIZE THE MACHINE
INITIALIZE THE TURING PLATFORM
T READS THE STORAGE CELL AT ITS FINITE CONTROL
T WRITES READ CELL VALUE TO REGISTER READABLE BY M
WRITES TO REMAINING INPUT OF M
EXECUTES ONE COMPUTATION STEP FOR M
WRITES SOME OUTPUT OF M TO REGISTER READABLE BY T
WRITES M'S COMPUTED DIRECTION OF MOVEMENT FOR T'S FINITE CONTROL TO A REMAINDER OF REGISTER READABLE BY T
READS THE REST OF M'S OUTPUT
T READS FROM THE REGISTERS READABLE TO T
T WRITES A NEW VALUE TO THE STORAGE CELL AT ITS FINITE CONTROL
T MOVES ITS FINITE CONTROL ONE CELL TO THE LEFT OR TO THE RIGHT OF OR NOT AT ALL
HAS HALTING CONDITION OCCURED?
NO
YES
FINISHED

FIG. 65



SPECIFY THE INITIAL VALUES OF $\vec{R}_1, \ldots, \vec{R}_m, \vec{C}, \vec{D}$
SPECIFY THE ELEMENTS IN P
SPECIFY INITIAL VALUES FOR ANY STORAGE CELLS?
YES
NO
SPECIFY VALUES FOR ONE OR MORE STORAGE CELLS $S_{\vec{P}}$ IN S
COMPUTING THE VALUES $P_{\vec{C}}$ AND $S_{\vec{D}}$
FINISHED

FIG. 66



INITIALIZE THE REGISTER MACHINE
COMPUTE NEXT INSTRUCTION POINTER $\vec{C}'$
COMPUTE NEXT STORAGE POINTER $\vec{D}'$
COMPUTE NEW VALUE $\vec{S}'$ TO BE WRITTEN TO CELL $\vec{D}'$
COMPUTE REGISTER TRANSITION MAPPING
SET $S_{\vec{D}} = \vec{S}'$, $\vec{C} = \vec{C}'$, $\vec{D} = \vec{D}'$
COMPUTE $P_{\vec{C}}$ AND $S_{\vec{D}}$ USING INPUT, IF ANY, FROM HOST
HAS A HALTING CONDITION OCCURED?
YES
FINISHED

FIG. 67

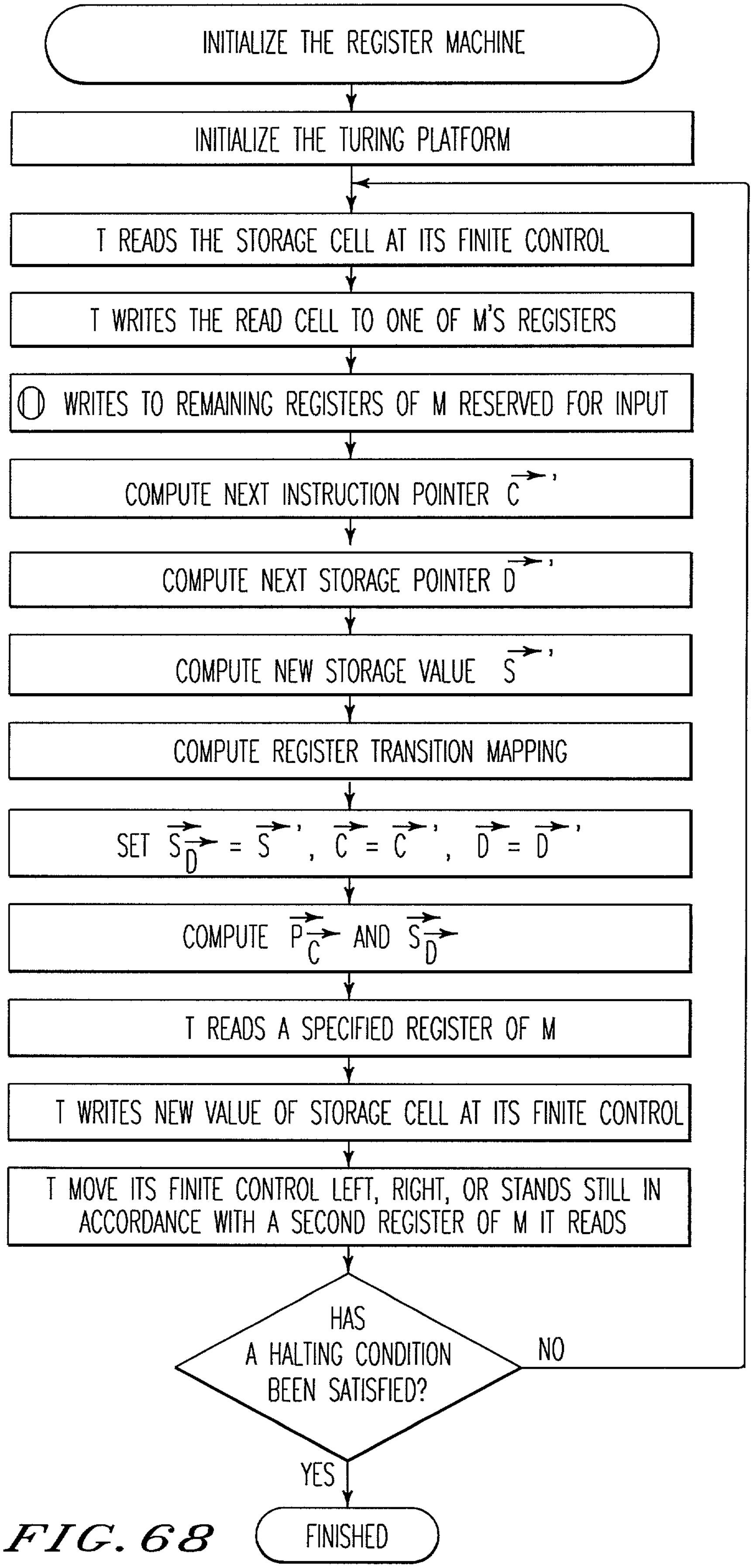

INITIALIZE THE REGISTER MACHINE
INITIALIZE THE TURING PLATFORM
T READS THE STORAGE CELL AT ITS FINITE CONTROL
T WRITES THE READ CELL TO ONE OF M'S REGISTERS
WRITES TO REMAINING REGISTERS OF M RESERVED FOR INPUT
COMPUTE NEXT INSTRUCTION POINTER $\vec{C}'$
COMPUTE NEXT STORAGE POINTER $\vec{D}'$
COMPUTE NEW STORAGE VALUE $\vec{S}'$
COMPUTE REGISTER TRANSITION MAPPING
SET $\vec{S}_{\vec{D}} = \vec{S}'$, $\vec{C} = \vec{C}'$, $\vec{D} = \vec{D}'$
COMPUTE $\vec{P}_{\vec{C}}$ AND $\vec{S}_{\vec{D}}$
T READS A SPECIFIED REGISTER OF M
T WRITES NEW VALUE OF STORAGE CELL AT ITS FINITE CONTROL
T MOVE ITS FINITE CONTROL LEFT, RIGHT, OR STANDS STILL IN ACCORDANCE WITH A SECOND REGISTER OF M IT READS
HAS A HALTING CONDITION BEEN SATISFIED?
NO
YES
FINISHED

FIG. 68

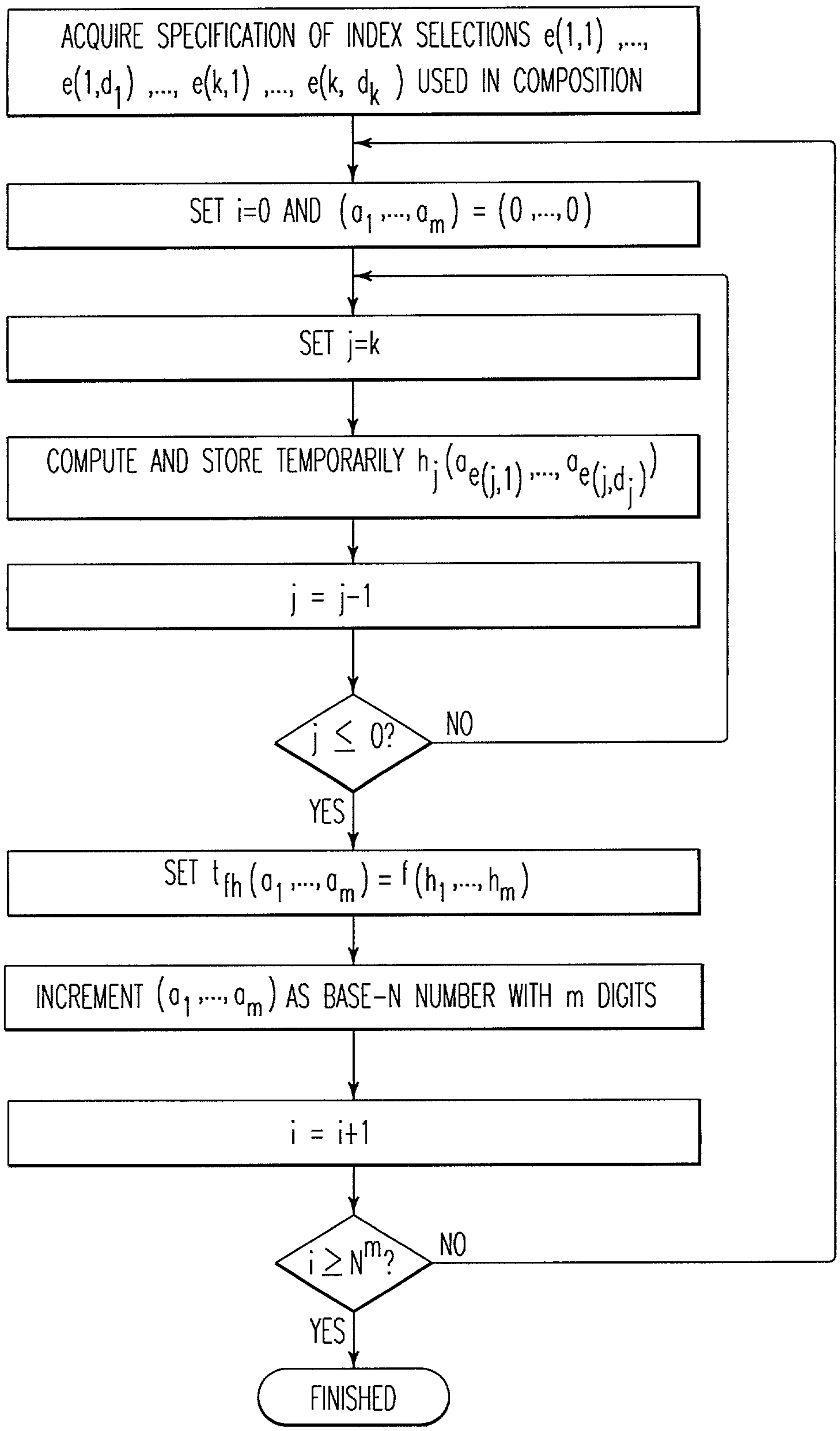

ACQUIRE SPECIFICATION OF INDEX SELECTIONS e(1,1) ,..., e(1,d1) ,..., e(k,1) ,..., e(k, dk) USED IN COMPOSITION
SET i=0 AND (a1 ,..., am) = (0 ,..., 0)
SET j=k
COMPUTE AND STORE TEMPORARILY hj (ae(j,1) ,..., ae(j,dj))
j = j-1
j ≤ 0?
NO
YES
SET tfh (a1 ,..., am) = f (h1 ,..., hm)
INCREMENT (a1 ,..., am) AS BASE-N NUMBER WITH m DIGITS
i = i+1
i ≥ N^m?
NO
YES
FINISHED

FIG. 69

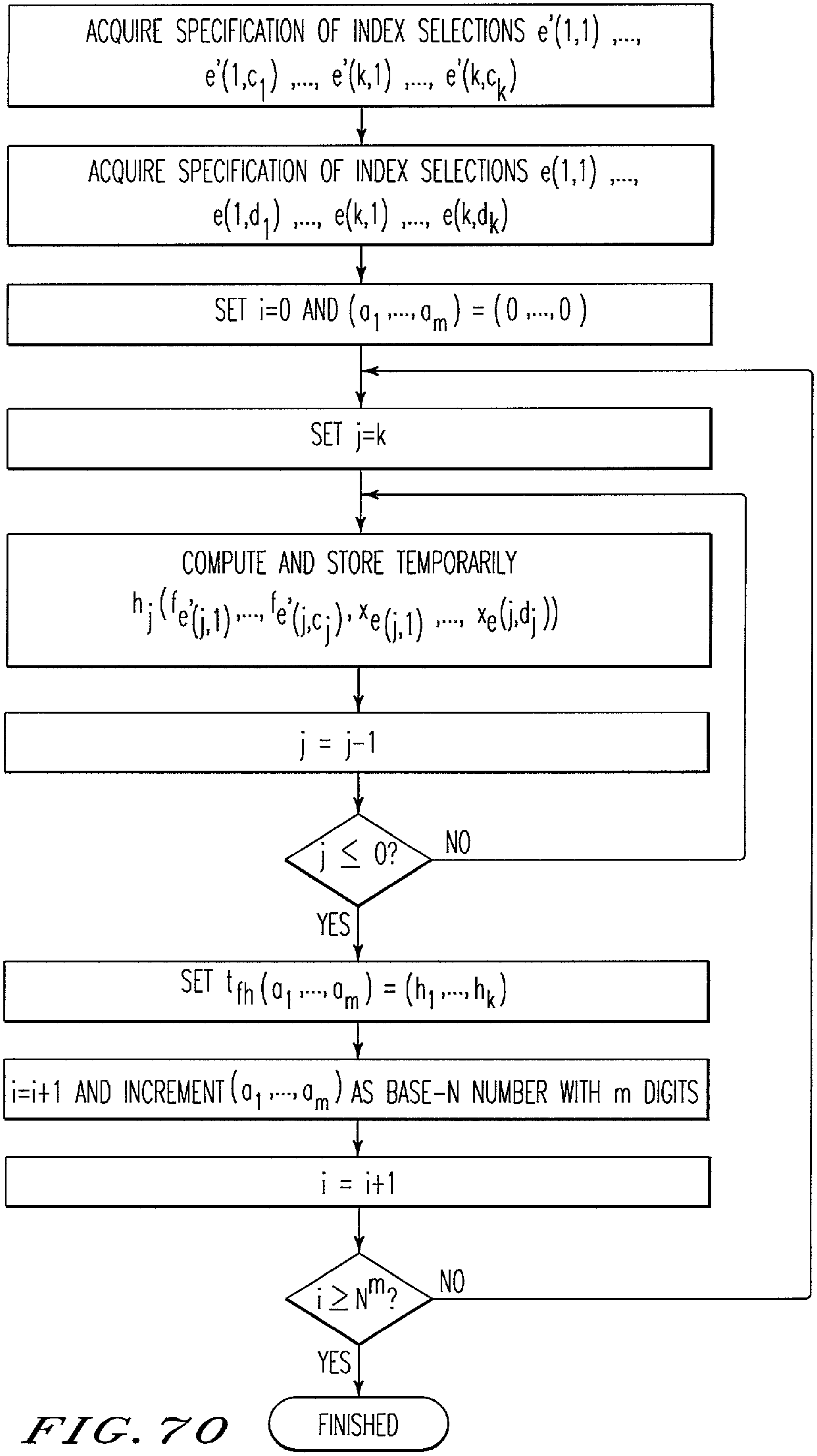

ACQUIRE SPECIFICATION OF INDEX SELECTIONS e'(1,1) ,..., e'(1,c1) ,..., e'(k,1) ,..., e'(k,ck)
ACQUIRE SPECIFICATION OF INDEX SELECTIONS e(1,1) ,..., e(1,d1) ,..., e(k,1) ,..., e(k,dk)
SET i=0 AND (a1 ,..., am) = (0 ,..., 0)
SET j=k
COMPUTE AND STORE TEMPORARILY
hj(fe'(j,1) ,..., fe'(j,cj), xe(j,1) ,..., xe(j,dj))
j = j-1
j ≤ 0?
NO
YES
SET tfh(a1 ,..., am) = (h1 ,..., hk)
i=i+1 AND INCREMENT (a1 ,..., am) AS BASE-N NUMBER WITH m DIGITS
i = i+1
i ≥ N^m?
NO
YES
FINISHED

FIG. 70

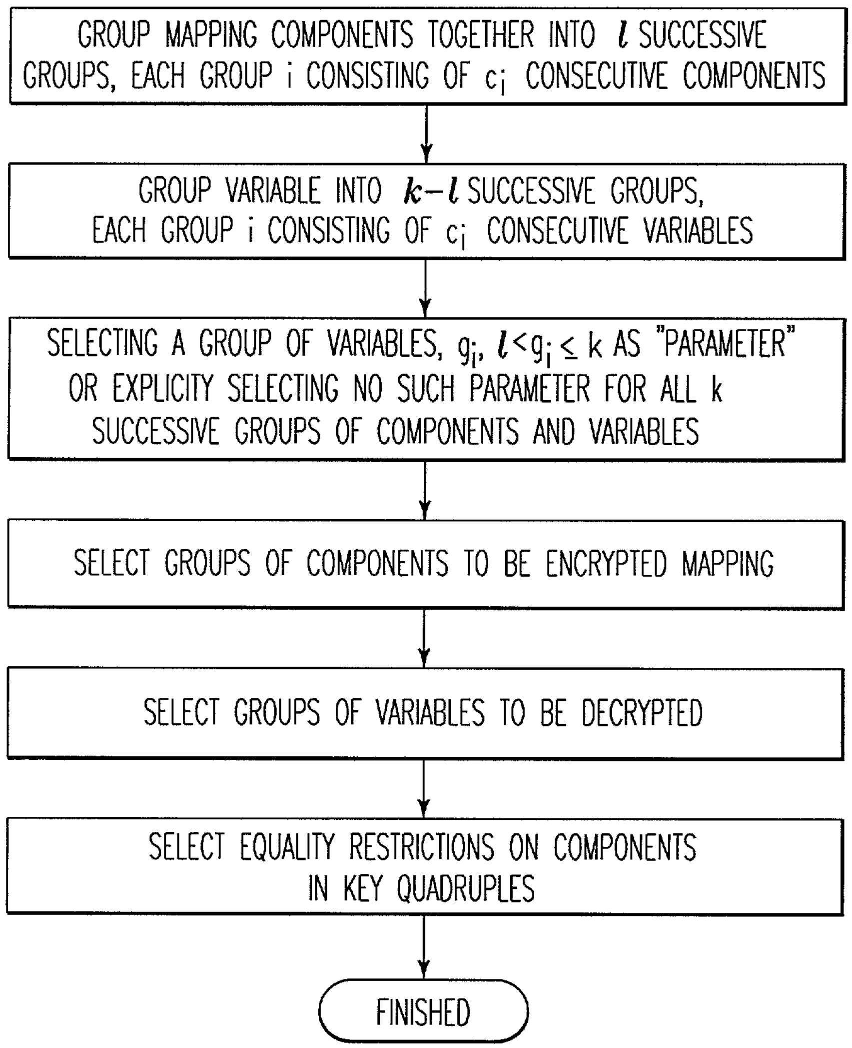

GROUP MAPPING COMPONENTS TOGETHER INTO l SUCCESSIVE GROUPS, EACH GROUP i CONSISTING OF c_i CONSECUTIVE COMPONENTS
GROUP VARIABLE INTO $k-l$ SUCCESSIVE GROUPS, EACH GROUP i CONSISTING OF c_i CONSECUTIVE VARIABLES
SELECTING A GROUP OF VARIABLES, g_i, $l<g_i \leq k$ AS "PARAMETER" OR EXPLICITY SELECTING NO SUCH PARAMETER FOR ALL k SUCCESSIVE GROUPS OF COMPONENTS AND VARIABLES
SELECT GROUPS OF COMPONENTS TO BE ENCRYPTED MAPPING
SELECT GROUPS OF VARIABLES TO BE DECRYPTED
SELECT EQUALITY RESTRICTIONS ON COMPONENTS IN KEY QUADRUPLES
FINISHED

FIG. 71

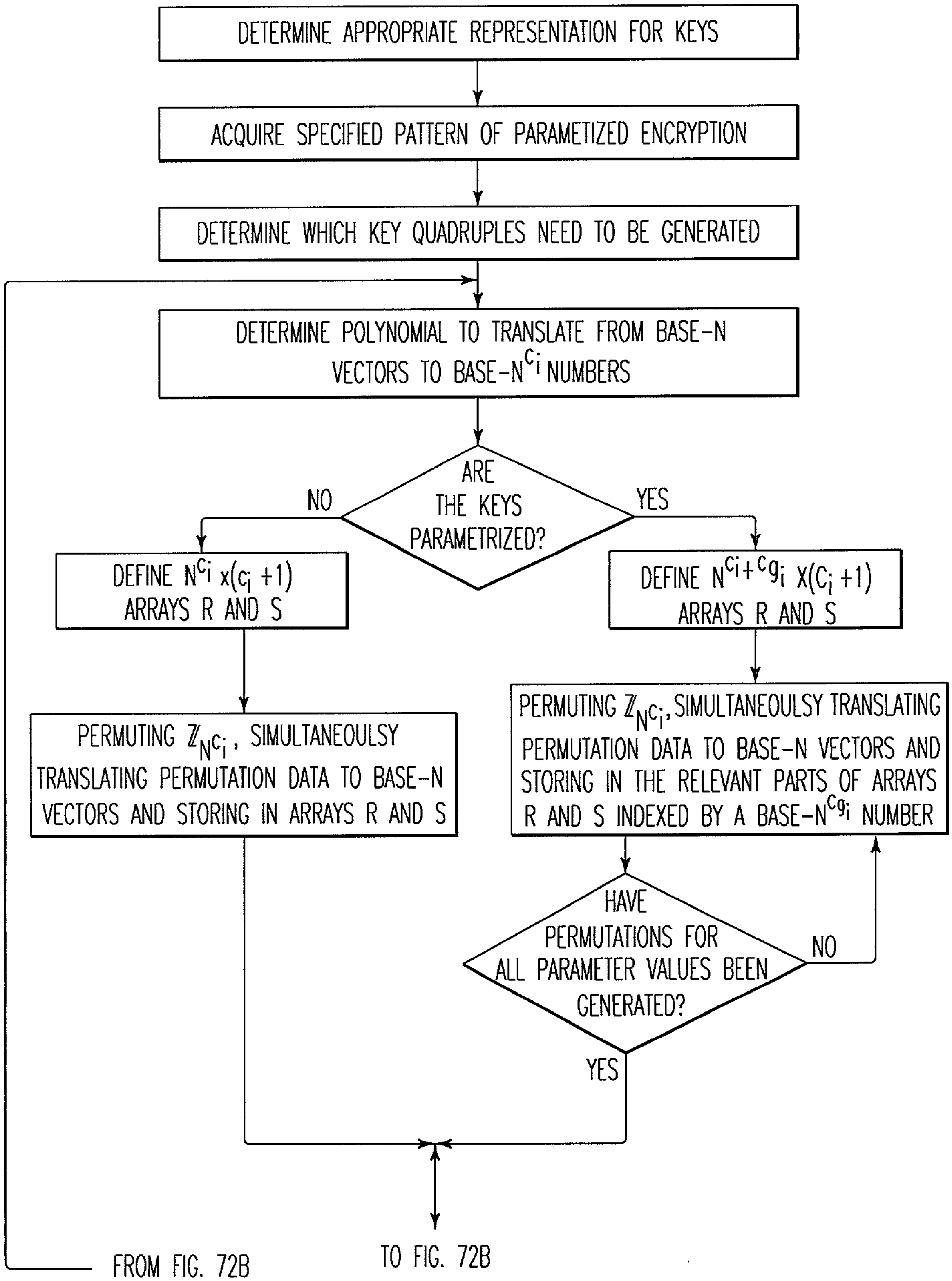

DETERMINE APPROPRIATE REPRESENTATION FOR KEYS
ACQUIRE SPECIFIED PATTERN OF PARAMETIZED ENCRYPTION
DETERMINE WHICH KEY QUADRUPLES NEED TO BE GENERATED
DETERMINE POLYNOMIAL TO TRANSLATE FROM BASE-N VECTORS TO BASE-N^{c_i} NUMBERS
ARE THE KEYS PARAMETRIZED?
NO
YES
DEFINE $N^{c_i} \times (c_i + 1)$ ARRAYS R AND S
DEFINE $N^{c_i + c_{g_i}} \times (C_i + 1)$ ARRAYS R AND S
PERMUTING $\mathbb{Z}_{N^{c_i}}$, SIMULTANEOULSY TRANSLATING PERMUTATION DATA TO BASE-N VECTORS AND STORING IN ARRAYS R AND S
PERMUTING $\mathbb{Z}_{N^{c_i}}$, SIMULTANEOULSY TRANSLATING PERMUTATION DATA TO BASE-N VECTORS AND STORING IN THE RELEVANT PARTS OF ARRAYS R AND S INDEXED BY A BASE-$N^{c_{g_i}}$ NUMBER
HAVE PERMUTATIONS FOR ALL PARAMETER VALUES BEEN GENERATED?
NO
YES
FROM FIG. 72B
TO FIG. 72B

FIG. 72A

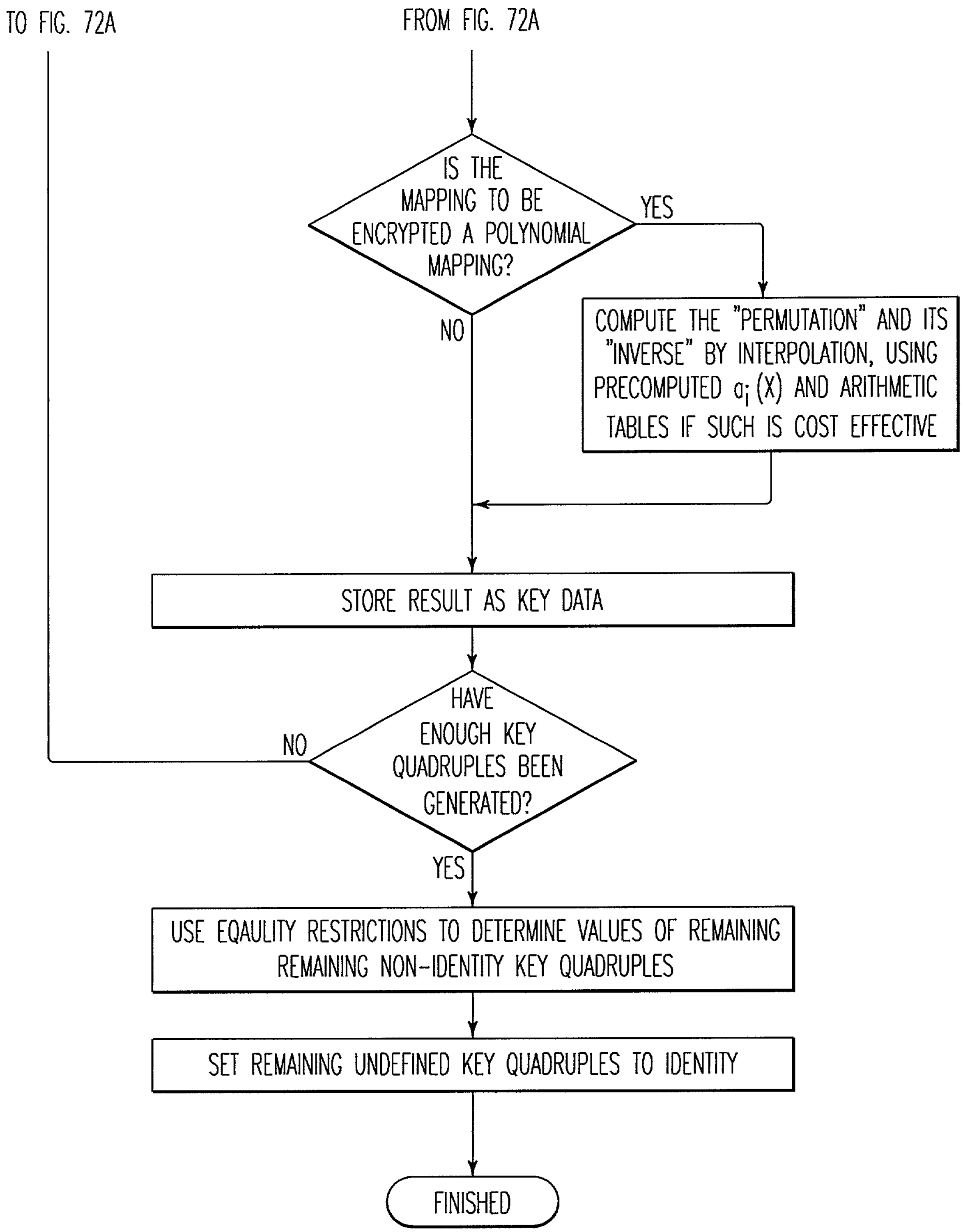

TO FIG. 72A
FROM FIG. 72A
IS THE MAPPING TO BE ENCRYPTED A POLYNOMIAL MAPPING?
YES
COMPUTE THE "PERMUTATION" AND ITS "INVERSE" BY INTERPOLATION, USING PRECOMPUTED $a_i(X)$ AND ARITHMETIC TABLES IF SUCH IS COST EFFECTIVE
NO
STORE RESULT AS KEY DATA
HAVE ENOUGH KEY QUADRUPLES BEEN GENERATED?
NO
YES
USE EQAULITY RESTRICTIONS TO DETERMINE VALUES OF REMAINING REMAINING NON-IDENTITY KEY QUADRUPLES
SET REMAINING UNDEFINED KEY QUADRUPLES TO IDENTITY
FINISHED

FIG. 72B

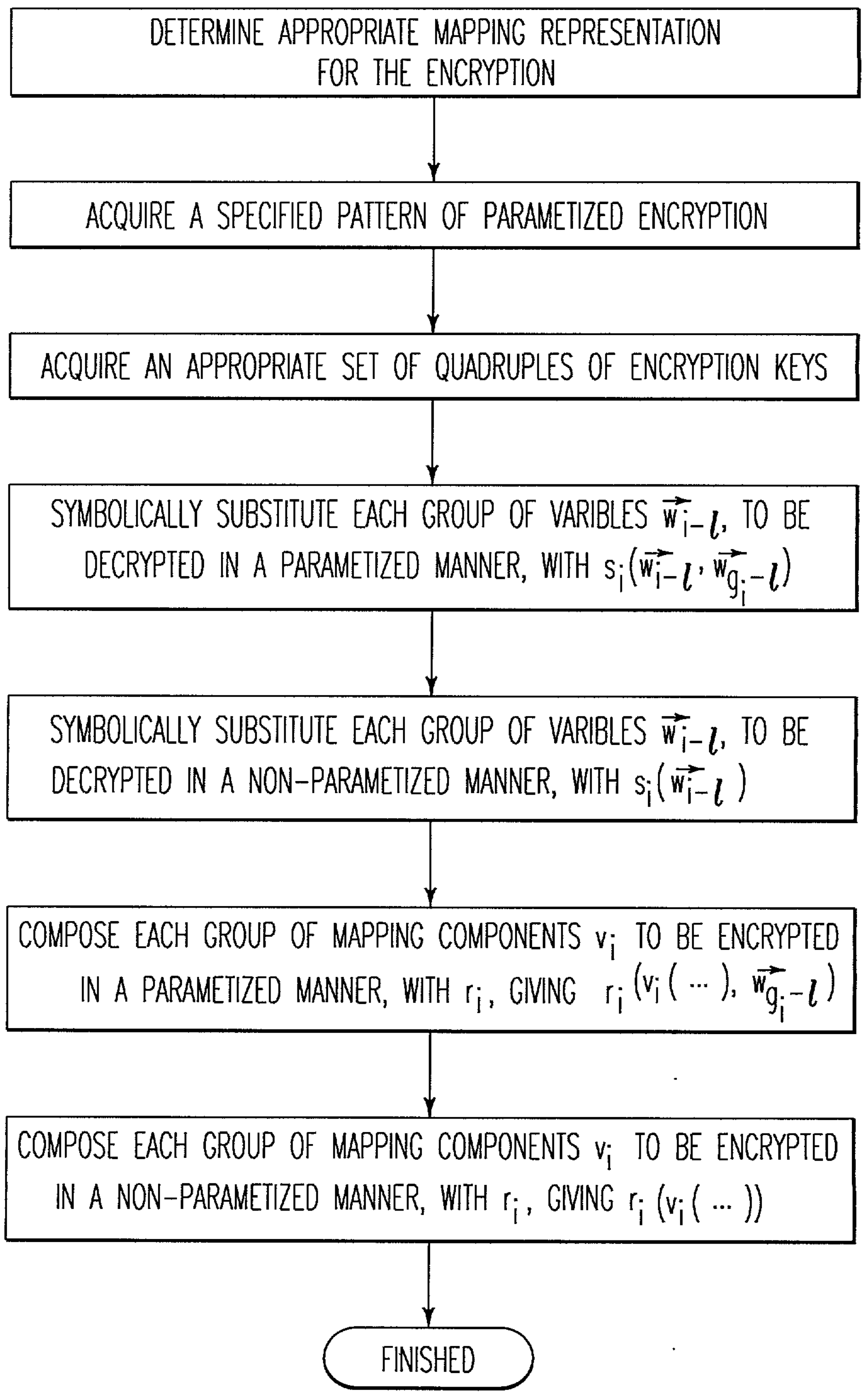

DETERMINE APPROPRIATE MAPPING REPRESENTATION FOR THE ENCRYPTION
ACQUIRE A SPECIFIED PATTERN OF PARAMETIZED ENCRYPTION
ACQUIRE AN APPROPRIATE SET OF QUADRUPLES OF ENCRYPTION KEYS
SYMBOLICALLY SUBSTITUTE EACH GROUP OF VARIBLES $\vec{w}_i - l$, TO BE DECRYPTED IN A PARAMETIZED MANNER, WITH $s_i(\vec{w}_i - l, \vec{w}_{g_i} - l)$
SYMBOLICALLY SUBSTITUTE EACH GROUP OF VARIBLES $\vec{w}_i - l$, TO BE DECRYPTED IN A NON-PARAMETIZED MANNER, WITH $s_i(\vec{w}_i - l)$
COMPOSE EACH GROUP OF MAPPING COMPONENTS v_i TO BE ENCRYPTED IN A PARAMETIZED MANNER, WITH r_i, GIVING $r_i(v_i(\dots), \vec{w}_{g_i} - l)$
COMPOSE EACH GROUP OF MAPPING COMPONENTS v_i TO BE ENCRYPTED IN A NON-PARAMETIZED MANNER, WITH r_i, GIVING $r_i(v_i(\dots))$
FINISHED

FIG. 73

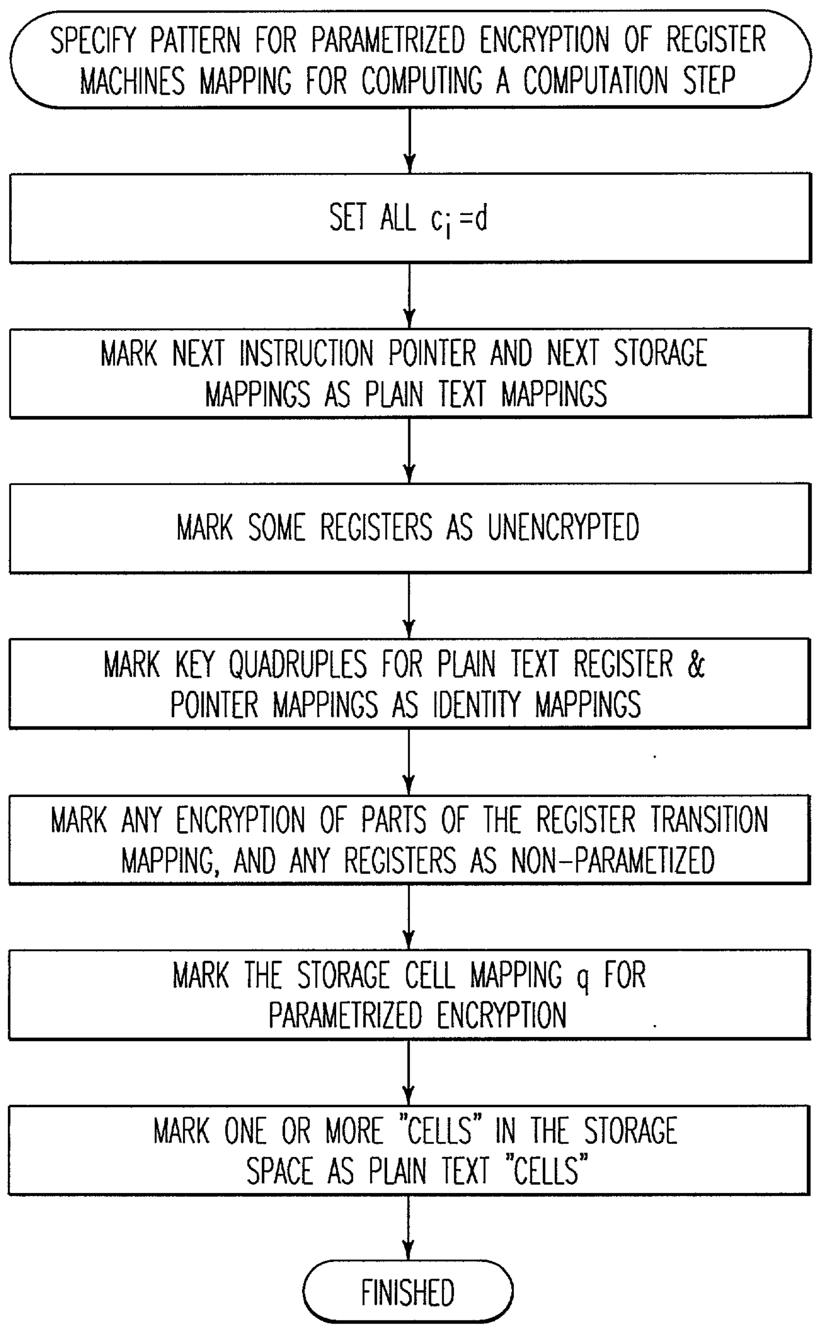

SPECIFY PATTERN FOR PARAMETRIZED ENCRYPTION OF REGISTER MACHINES MAPPING FOR COMPUTING A COMPUTATION STEP
SET ALL c_i =d
MARK NEXT INSTRUCTION POINTER AND NEXT STORAGE MAPPINGS AS PLAIN TEXT MAPPINGS
MARK SOME REGISTERS AS UNENCRYPTED
MARK KEY QUADRUPLES FOR PLAIN TEXT REGISTER & POINTER MAPPINGS AS IDENTITY MAPPINGS
MARK ANY ENCRYPTION OF PARTS OF THE REGISTER TRANSITION MAPPING, AND ANY REGISTERS AS NON-PARAMETIZED
MARK THE STORAGE CELL MAPPING q FOR PARAMETRIZED ENCRYPTION
MARK ONE OR MORE "CELLS" IN THE STORAGE SPACE AS PLAIN TEXT "CELLS"
FINISHED

FIG. 74

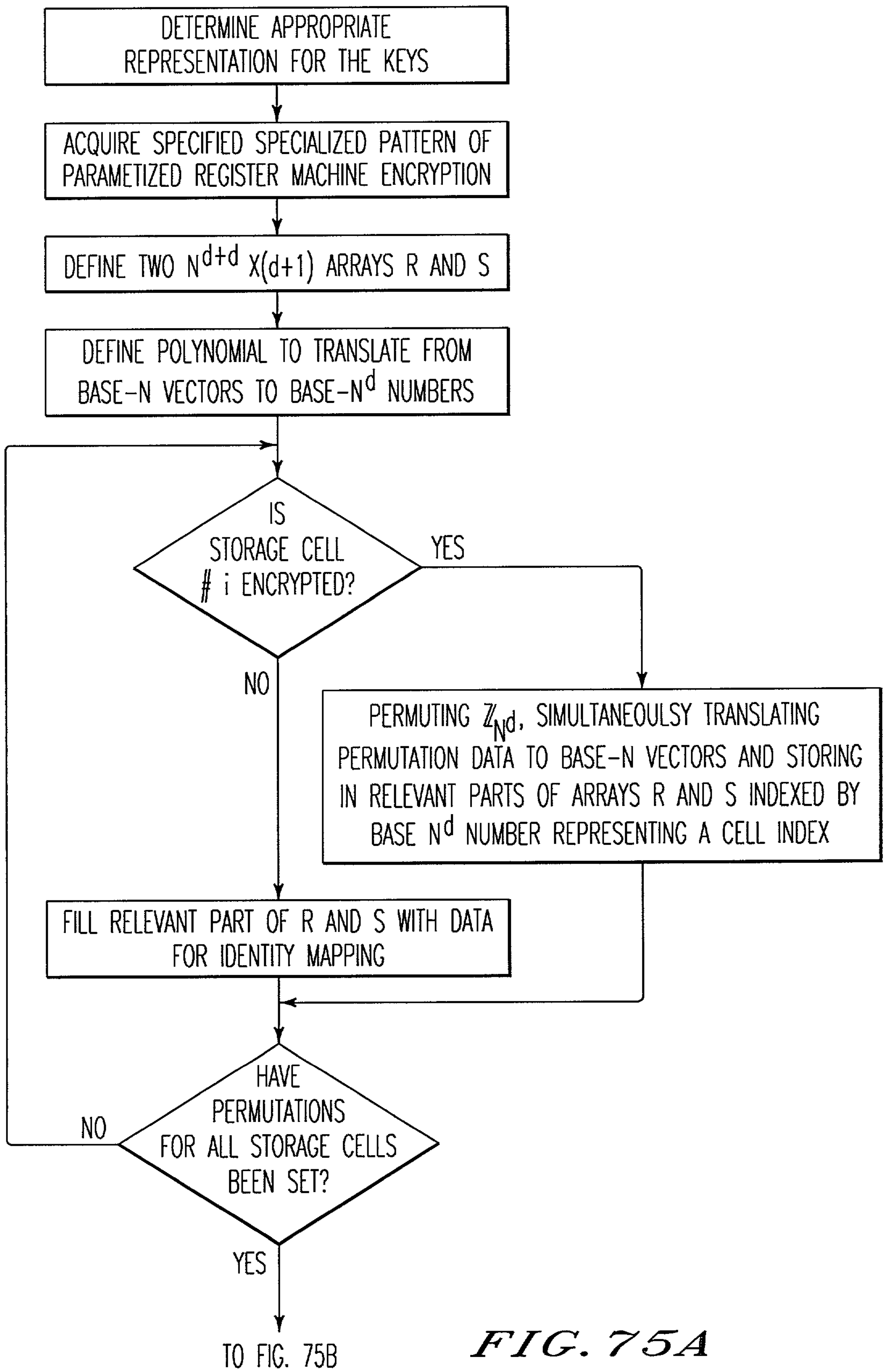

DETERMINE APPROPRIATE REPRESENTATION FOR THE KEYS
ACQUIRE SPECIFIED SPECIALIZED PATTERN OF PARAMETIZED REGISTER MACHINE ENCRYPTION
DEFINE TWO N^d+d X(d+1) ARRAYS R AND S
DEFINE POLYNOMIAL TO TRANSLATE FROM BASE-N VECTORS TO BASE-N^d NUMBERS
IS STORAGE CELL # i ENCRYPTED?
YES
NO
PERMUTING Z_N^d, SIMULTANEOULSY TRANSLATING PERMUTATION DATA TO BASE-N VECTORS AND STORING IN RELEVANT PARTS OF ARRAYS R AND S INDEXED BY BASE N^d NUMBER REPRESENTING A CELL INDEX
FILL RELEVANT PART OF R AND S WITH DATA FOR IDENTITY MAPPING
HAVE PERMUTATIONS FOR ALL STORAGE CELLS BEEN SET?
NO
YES
TO FIG. 75B

FIG. 75A

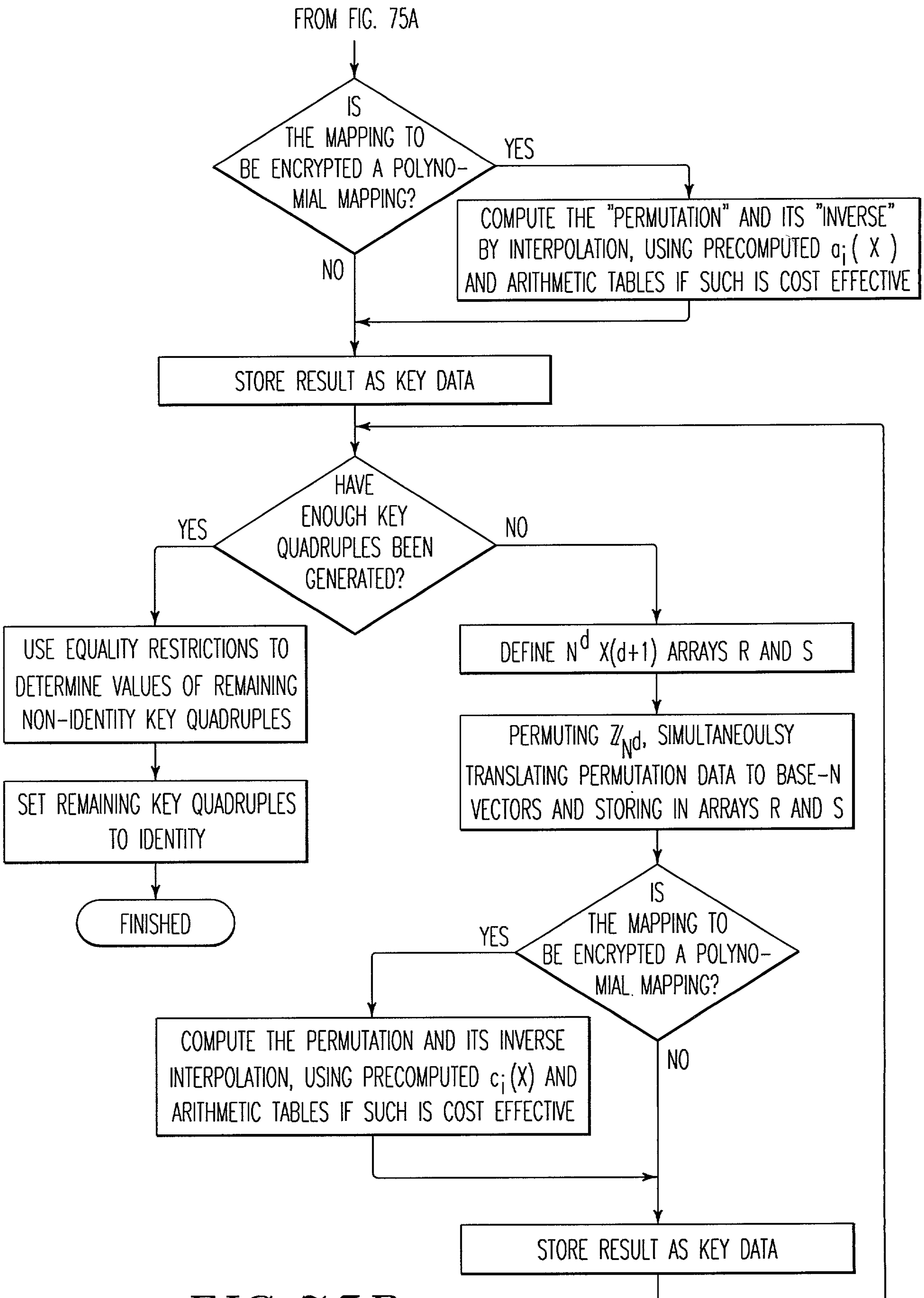

FROM FIG. 75A
IS THE MAPPING TO BE ENCRYPTED A POLYNOMIAL MAPPING?
YES
COMPUTE THE "PERMUTATION" AND ITS "INVERSE" BY INTERPOLATION, USING PRECOMPUTED $a_i(X)$ AND ARITHMETIC TABLES IF SUCH IS COST EFFECTIVE
NO
STORE RESULT AS KEY DATA
HAVE ENOUGH KEY QUADRUPLES BEEN GENERATED?
YES
NO
USE EQUALITY RESTRICTIONS TO DETERMINE VALUES OF REMAINING NON-IDENTITY KEY QUADRUPLES
SET REMAINING KEY QUADRUPLES TO IDENTITY
FINISHED
DEFINE N^d X(d+1) ARRAYS R AND S
PERMUTING $\mathbb{Z}_{N^d}$, SIMULTANEOULSY TRANSLATING PERMUTATION DATA TO BASE-N VECTORS AND STORING IN ARRAYS R AND S
IS THE MAPPING TO BE ENCRYPTED A POLYNOMIAL MAPPING?
YES
NO
COMPUTE THE PERMUTATION AND ITS INVERSE INTERPOLATION, USING PRECOMPUTED $c_i(X)$ AND ARITHMETIC TABLES IF SUCH IS COST EFFECTIVE
STORE RESULT AS KEY DATA

FIG. 75B

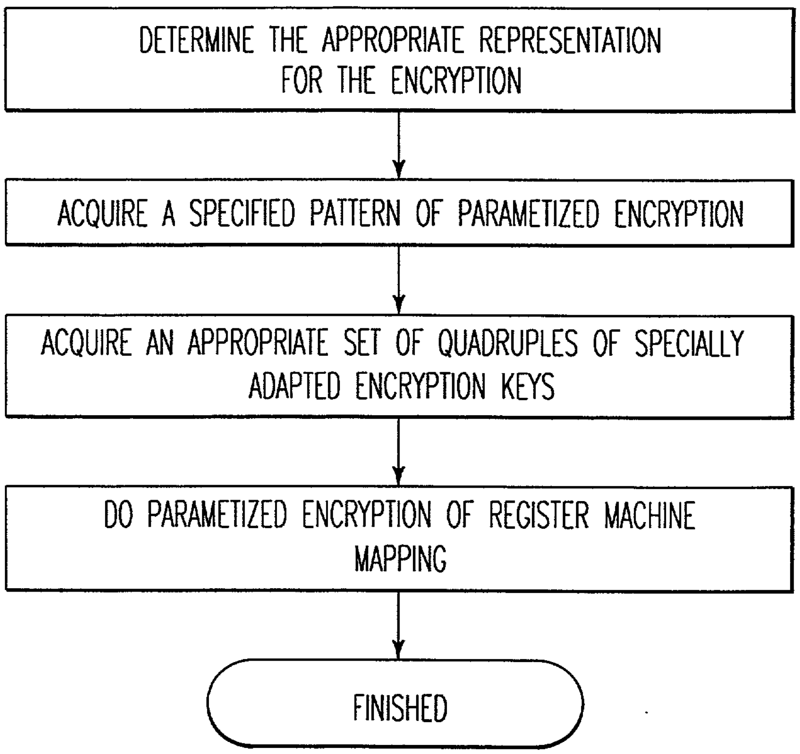

DETERMINE THE APPROPRIATE REPRESENTATION FOR THE ENCRYPTION
ACQUIRE A SPECIFIED PATTERN OF PARAMETIZED ENCRYPTION
ACQUIRE AN APPROPRIATE SET OF QUADRUPLES OF SPECIALLY ADAPTED ENCRYPTION KEYS
DO PARAMETIZED ENCRYPTION OF REGISTER MACHINE MAPPING
FINISHED

FIG. 76